All Source Sensor Integration Using an Extended Kalman Filter

THESIS

Timothy R. Penn, Second Lieutenant, USAF

AFIT/GE/ENG/12-32

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/12-32

All Source Sensor Integration Using an Extended Kalman Filter

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Timothy R. Penn, B.S.E.E.

Second Lieutenant, USAF

March 2012

AFIT/GE/ENG/12-32

All Source Sensor Integration Using an Extended Kalman Filter

Timothy R. Penn, B.S.E.E.
Second Lieutenant, USAF

Approved:

Dr. John K. Raquet (Chairman)	date
Maj Kenneth Fisher, PhD (Member)	date
Dr. Meir Pachter (Member)	date

Abstract

The global positioning system (GPS) has become an ubiquitous source for navigation in the modern age, especially since the removal of selective availability at the beginning of this century. The utility of the GPS is unmatched, however GPS is not available in all environments. Heavy reliance on GPS for navigation makes the warfighter increasingly vulnerability as modern warfare continues to evolve.

This research provides a method for incorporating measurements from a massive variety of sensors to mitigate GPS dependence. The result is the integration of sensor sets that encompass those examined in recent literature as well as some custom navigation devices. A full-state extended Kalman filter is developed and implemented, accommodating the requirements of the varied sensor sets and scenarios. Some 19 types of sensors are used in multiple quantities including inertial measurement units, single cameras and stereo pairs, 2D and 3D laser scanners, altimeters, 3-axis magnetometers, heading sensors, inclinometers, a stop sign sensor, an odometer, a step sensor, a ranging device, a signal of opportunity sensor, global navigation satellite system sensors, an air data computer, and radio frequency identification devices. Simulation data for all sensors was generated to test filter performance. Additionally, real data was collected and processed from an aircraft, ground vehicles, and a pedestrian.

Measurement equations are developed to relate sensor measurements to the navigation states. Each sensor measurement is incorporated into the filter using the Kalman filter measurement update equations. Measurement types are segregated based on whether they observe instantaneous or accumulated state information. Accumulated state measurements are incorporated using delayed-state update equations. All other measurements are incorporated using the numerically robust UD update equations. Simulation results show the expected performance of improved navigation state estimation over time with the systematic addition of each sensor.

Acknowledgements

I would like to thank my advisor for his invaluable aid and guidance throughout this project. Most of all, I wish to thank my beautiful little girls and my wonderful wife who supported me more than they can ever imagine. Thank you my little family for being my source of life and driving motivation throughout all times no matter the hardship. And finally, thanks to my little sister who promised to come to my graduation, but was called home too early to make it. I miss you more than you could ever know.

Timothy R. Penn

Contents

List of Abbreviations

Abbreviation		Page
GPS	Global Positioning System	1
ASPN	All Source Positioning Navigation	2
DARPA	Defense Advanced Research Agency	2
IMU	Inertial Measurement Unit	3
RFID	Radio Frequency Identification	3
EKF	Extended Kalman Filter	4
WGS-84	World Geodetic System 1984	7
ECEF	Earth Centered Earth Fixed frame	8
NED	North East Down	8
COG	Center of Gravity	8
DCM	Direction Cosine Matrix	11
TCB	Time Correlated Bias	18
WGN	White Gaussian Noise	18
PDF	Probability Density Function	19
UD	Upper Triangular Diagonal	27
UT	Unscented Transform	28
MEMS	Micro Electro-Mechanical Systems Agency	30
VIMU	Virtual Inertial Measurement Unit	30
LADAR	Laser Radar	31
LMS	Least Mean Squares	31
DOP	Dilution of Precision	31
WLMS	Weighted Least Mean Squares	32
LiDAR	Light Detection and Ranging	32
ADC	Air Data Computer	51
INS	Inertial Navigation System	64

Abbreviation		Page
GNSS	Global Navigation Satellite System	70
TRA	Terrain-Referenced Altimeter Identification	93
DTED	Digital Terrain Elevation Data	93
STD	Standard Deviation	96
UKF	Unscented Kalman Filter	136

All Source Sensor Integration Using an Extended Kalman Filter

I. Introduction

The advent of the Global Positioning System (GPS) ushered in a new era of navigation. Never before could a platform pin point its exact location on Earth with such accuracy and precision. Because of the advantages it offers, the Department of Defense has become extremely reliant on its use. Under many situations, the performance of GPS can be degraded or blocked entirely. This includes degradation by environment, terrain, jamming, and even the possibility for malicious misinformation attacks. January of 2010, the Chief of Staff of the Air Force delivered a warning stating our dependence on GPS makes us vulnerable, and that modern warfare will only continue to require higher precision navigation abilities [12]. Numerous research efforts have been undertaken to meet the evolving navigation requirements while lessening dependence on GPS. Although, no current technology can provide the equivalent navigation reliability of an undenied GPS constellation over time, the overall goal is to maintain high precision navigation information as accurately as possible for as long as it takes to regain quality GPS.

This thesis combines multiple previous methods to navigate using sensors in GPS-limited and denied environments. Positioning sensors of all types in multiple quantities are used to maintain accurate positioning information. Sensors were chosen, to encompass and combine the majority navigation measurement types employed in recent literature as well as to examine some custom devices. To the author's knowledge, no other previous research project can match this project for the number of types of sensors used in multiple quantities.

This research focuses primarily on the measurements and their relations to the navigation states. Measurement equations are developed for sensor preprocessed mea-

surements, and these navigation equations are not dependent upon the integrating filter. That is to say, the measurements were combined using an extended Kalman filter, but the same measurement equations could be adapted to any other filter such as the unscented Kalman filter, particle filter, etc.

1.1 All Source Positioning Navigation

The All Source Positioning Navigation (ASPN) program is headed by the Defense Advanced Research Agency (DARPA) and is one such effort to mitigate dependence on GPS. For the ASPN program, a standardized set of data from common navigation sensors in multiple quantities, on multiple platforms, and in multiple scenarios was collected. The ultimate goal of the program is to provide low-cost, robust, effective, navigation methods to the military operator in any environment regardless of GPS availability.

The role of this thesis in the ASPN program is to create standardized solution from the massive test bed of sensor data. At the time of the writing of this thesis, the real data was not available. Therefore, all data used in this thesis was simulated to represent the data that was to be received from the ASPN program.

1.2 Research Objectives

The main objective for this project is to provide an analysis of the effect of the fusion of multiple sensors on a platform's navigation solution. The integrity of the analysis is tested using simulation data. The effect on the navigation solution is observed by viewing the systematic addition of each individual sensor in designed scenarios. Sensors that provide information that is not directly observable or whose information becomes difficult to view in the presence of more descriptive sensors, are showcased in specialized scenarios where their performance can be observed.

1.3 Scope

Due to the seer size of the this project, the scope for sensor integration is mostly limited to methods currently available in literature. These methods have been expounded upon where necessary. What makes this research unique is the culmination of all these methods into a single project. All measurement equations are written to be matched directly to real data.

In addition, ASPN constraints required only "textbook" solutions for the development of the standard navigation solution to include the implementation of an extended Kalman filter. Part of the ASPN methodology was that no prior information with regard to sensor sets is known for filter development. Therefore, development of the filter required extreme flexility and relatively simple design. Due to this constraint, all measurements, including inertial data, was integrated into the filter using the traditional Kalman filter update. Processing time was not part of the APSN constraints, therefore treating inertial data as measurements updates was appropriate for this project. Bringing in inertial data using measurement models over error state estimation greatly simplified the problem of including multi-INS of various grades, or no INS at all, in the navigation system.

Multiple quantities of sensors are examined to analyze the combined effects. Certain sensors, particulary IMUs, are examined in various measurement qualities including MEMs and tactical grade. Other sensors are used in different configurations composing new sensors. Examples include single cameras and stereo camera pairs which are treated as separate sensors, and an optical sensor that detects stop signs. The cumulative list of sensors examined in this project are of the type: inertial measurement unit (IMU), GNSS position/velocity sensor, GNSS pseudorange/delta-range sensor, monocular optical camera, odometer, barometric altimeter, step sensor, stop sign sensor, 3D laser scanner, 3-axis magnetometer, ranging device, 2D laser scanner, terrain-referenced altimeter, inclinometer, magnetic compass, radio frequency identification (RFID), signal of opportunity, air data computer, and stereo optical camera.

Sensor measurement updates in this thesis are performed independent of other sensor measurements; however, some of these measurements were preprocessed using information from other sensors. Measurement independence however is preserved as alternate sensor aiding uses the measurements indirectly to speed processing and increase accuracy.

1.4 Thesis Overview

Chapter II of this thesis contains the mathematical background necessary to complete this work as well as mathematical notation description. This includes detailed descriptions of the conventional Kalman filter, extended Kalman filter (EKF), Kalman filter delayed state equations, coordinate frames and coordinate frame transformations. Also in this chapter is a discussion of previous research projects over topics that relate to this thesis.

In Chapter III, each sensor that was used in the project is discussed. This includes measurement models, Kalman filter measurement update equations, incorporation of lever arms, and sensor error states specifics. The main filter coding algorithm is explained as well in this chapter, since one of the main difficulties of this thesis was simply handling the plug and play nature of the massive sensor test bed. This includes no assumptions with regard to the sensors in operation during a run, number of states being estimated, or filter run time.

In Chapter IV the results of this project are presented with comparison between number of sensors used and accuracy. Sensors are added systematically for various scenarios and the solution error and covariance are discussed. This is performed for simulation data only as the real data for the project did meet the time constraints on the delivery of this thesis.

Chapter V is a summary of the previous chapters of this thesis. In the thesis summary, filter performance and future work is discussed. An overview of the overall

code implementation is discussed as well as implementations and methods that worked well and what would be done differently if this thesis were to be done over again.

II. Background

This chapter outlines the background material required to complete this thesis. Required reference frames for navigation and the translations between them are discussed in detail. The preferred attitude representation representation method is chosen as well as the conversion between other methods which is useful for many of the varied measurement set that will later be analyzed in chapter III. Finally a brief overview of Kalman filter theory is given a with some deviation from the standard implementations.

2.1 Mathematical Notation

The mathematical notation in this document is laid out in this section. Variable names and annotations have been chosen to be both intuitive and meaningful wherever possible.

2.1.1 Dimension. The navigation in this thesis is carried in three dimensional space and the following notation is used to differentiate quantity dimension. Scalars are signified by a lower or upper-case case variables (e.g., x). Vectors are by default given in column form and represented using lower-case bold letters (e.g., $\mathbf{x}$). Matrices are represented using uppercase bolded letters (e.g., $\mathbf{X}$).

2.1.2 Homogeneous Pointing Vectors. Homogenous pointing vectors are defined to have 1 for the last element (i.e., all elements divided by the z dimension) [31]. These quantities are denoted by an under bar (e.g., $\underline{\mathbf{x}}$). They are associated with the native camera measurement in this document.

2.1.3 Other. Physical quantities have associated errors and will be represented by a tilde or hat accent (e.g., $\tilde{x}$ or $\hat{x}$). Theoretical/true quantities omit the accent. Unit vectors are identified using an over bar (e.g., $\|\bar{\mathbf{u}}\| = 1$).

2.2 WGS-84 Reference System

The 1984 World Geodetic System (WGS-84) [6] reference system models the shape of the earth as an ellipsoid. The ellipsoid shape is used due to the earth's bulging at the equator and flattening at the poles. The ellipsoid is defined to rotate about the z axis coincident with the polar axis. The position of any point p on the ellipsoid is described using a geodetic latitude (L), longitude (λ) and altitude (Alt). Geodetic latitude is defined by the angle of a line that passes through some point p normal to the surface of the ellipsoid with the equatorial plane. Now considering a plane that contains this line, a second ellipsoid is formed by the intersection of this plane and the ellipsoid. The radius of curvature of this new ellipsoid at point p is called the radius of curvature in the prime vertical [25] and is given by:

$$R_p = \frac{a}{(1 - e^2 \sin L)^{1/2}} \tag{2.1}$$

Where a is the semi-major axis and e is the eccentricity of the ellipsoid. The radius of curvature that fits the meridian at the latitude chosen is called the radius of curvature in the meridian and is given by:

$$R_m = \frac{a(1 - e^2)}{(1 - e^2 \sin L)^{3/2}} \tag{2.2}$$

The radius of curvature in the prime vertical and meridian are required to convert geodetic latitude and longitude to a local-level navigation frame, when taking into account the earth's ellipsoidal shape.

2.3 Reference Frames

This section includes reference frames designed for tracking the vehicle navigation states of position, velocity and attitude. Sensor frames are generically described in this section, however specific sensor frames are only discussed in Chapter III of this thesis, since there are simply too many to list in multiple locations.

2.3.1 Inertial Frame. The inertial reference frame, denoted by x^i, y^i and z^i in Figure 2.1, is the coordinate frame in which Newton's laws of motion are valid. This frame originates at the center of the earth with the vertical z axis passing through the North Pole. The x axis points in an arbitrarily chosen direction that remains fixed in orientation and together with the y axis completes the right handed Cartesian coordinate system.

2.3.2 Earth Frame. The earth centered earth fixed frame (ECEF) reference frame, denoted by x^e, y^e and z^e in Figure 2.1. is initially aligned with the inertial frame and rotates at the earth sidereal rate. The x axis projects through the prime meridian at the equator, and the y axis completes the three axis triad using the right handed convention. The rotation of the earth with respect to the inertial frame in skew symmetric form is described as:

$$\mathbf{\Omega}_{ie}^{e} = \begin{bmatrix} 0 & 0 & \omega_{ie_z}^{e} \end{bmatrix}^{T} \times \tag{2.3}$$

where $\omega_{ie_z}^{e}$ is the earth sideral rate about the polar axis and defined in the WGS-84 reference datum described in Section 2.2. The skew symmetric operator $\times$ acts on a 3 element column vector $\boldsymbol{\omega}$ as shown in (2.4) allowing a vector cross product to be represented by a matrix vector multiplication.

$$\boldsymbol{\omega}\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.4}$$

where R_p is the radius of curvature in the prime vertical defined in (2.1) and e is the eccentricity of the earth.

2.3.3 Navigation Frame. The north, east, and down (NED) frame, denoted by N, E and D in Figure 2.1, is commonly used to track the a platform. The NED frame is fixed at the platform's center of gravity (COG) throughout navigation. The

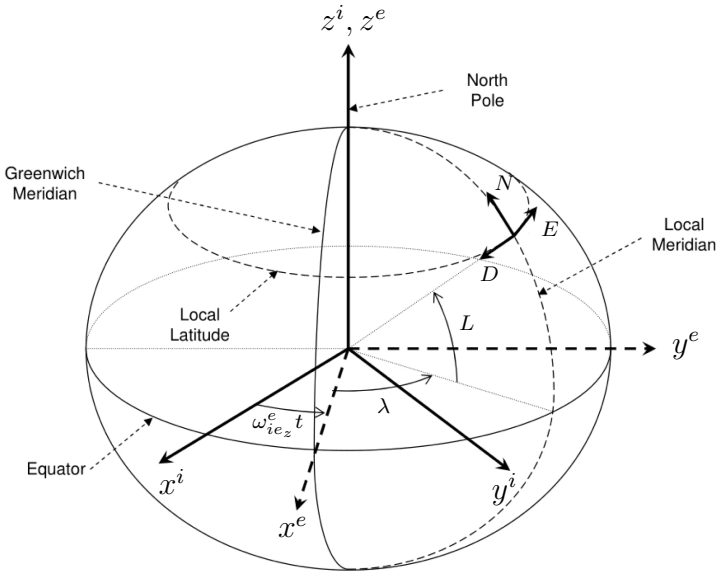

Figure 2.1: ECEF, inertial, and NED reference frames from [31]

frame axes are fixed pointing North, East, and Down along the lines of geodetic latitude, longitude, and altitude. Because of the curvature of the earth, the NED frame rotates with its earth surface displacement. This rotation rate is described in [29] as:

$$\mathbf{\Omega}^n_{en} = \begin{bmatrix} \frac{v_E}{(R_p+h)} & \frac{-v_N}{(R_m+h)} & \frac{-v_E \tan(L)}{(R_m+h)} \end{bmatrix}^T \times \tag{2.5}$$

Where v_E, v_N and v_D are platform velocities in the NED frame, R_p is the radius of curvature in the prime vertical given in (2.1), and R_m is the radius of curvature in the meridian given in (2.2).

The platform position is tracked using the geodetic navigation frame given by a geodetic latitude, longitude and altitude in units of *deg* or *rad*, *deg* or *rad*, and *m* respectively. This follows a north, east, and up convention. This frame is only used for the geodetic position. The geodetic frame is related to the NED frame by a change of units in the north and east channels, and a sign reversal on the vertical component.

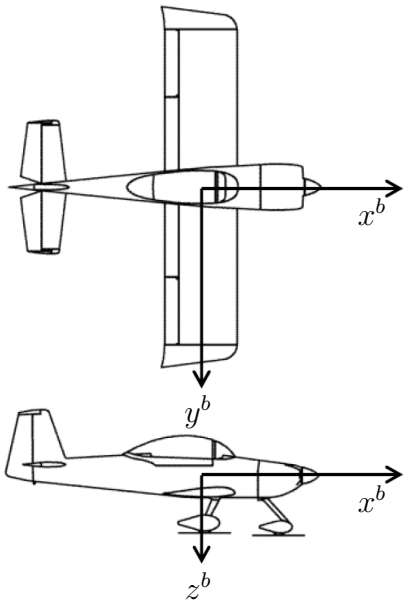

Figure 2.2: From [31], this figure demonstrates the body frame used in this thesis

2.3.4 Body Frame. Similar to the navigation frame, the body frame is located on the vehicle and the origin of the frame is placed at the location of the guidance system [29]. The body frame utility derives from multiple sources such as describing the platform dynamics, and many (if not most) sensors record measurements in the body frame. In order to navigate with body frame information, the body attitude must be known to relate it to the navigation frame.

2.4 Sensor Frames

Given the massive array of sensors used in the project, it is impossible to define a standard frame in which all the measurements are observed. As a result, transformations must be defined to convert between the sensor frame and the body frame. This is required show the relationship of the measurements to the navigation states. In this thesis, sensor frames where defined to match those seen commonly in literature and are given in the section where the sensor is discussed.

2.5 Attitude Representations

There are multiple methods of attitude representation in practice. The three types referenced in this document are direction cosines, Euler angles and quaternions. These three representations are discussed in the following subsections.

2.5.1 Direction Cosine Matrices. A direction cosine matrix (DCM) is a nine element rotational matrix. The rows of the body to navigation frame represent unit vectors in the body frame projected onto the reference axes [29]. Direction cosines have the benefit of a good physical interpretation, but require nine elements to describe which is significantly more than other representations.

2.5.2 Euler Angles. Euler angles represent the three angles (ϕ, θ and ψ) representing roll, pitch and yaw. Euler angles provide perhaps the most intuitive sense of attitude representation; however, they have several potential pitfalls regarding their use. Namely these pitfalls are the discontinuities inherent in trigonometric functions and high nonlinearity attributes.

2.5.3 Quaternions. Quaternions are four parameter normalized hyper complex quantities where the three complex components form an axis of rotation, and the real component describes the rotation about that axis. Quaternions are not characterized by the high nonlinearities experienced with Euler angles or gimbal lock. Their main drawback stems from the more obscure physical interpretation and addition of a fourth element compared with the Euler angles. Regardless of these drawbacks, it is generally agreed the advantages outweigh the disadvantages making them most often the implementation of choice in strapdown navigation systems [11].

Given the prior description of the three main attitude representations, the quaternion representation will be the method employed throughout this thesis to track the platform attitude. Transformations can be handled directly using the quaternion elements; however, the DCM will be used for body to NED transformations to be consistent with literature and because of the intuitive nature of the operation.

2.6 Coordinate Frame Transformations

Most coordinate transformations performed in this document are handled by DCMs. DCMs are very useful for transforming quantities between reference frames and simply performing rotations within the same reference frame. The general mechanics involved with the DCM rotations are described in this section as well some specific rotations between frames. Other relationships are also given for converting between frames when more than a rotation is required.

2.6.1 DCM Properties. DCM rotations, are accomplished using the DCM multiplication as:

$$\mathbf{v}^a = \mathbf{C}_b^a \mathbf{v}^b \tag{2.6}$$

In this example the vector $\mathbf{v}^b$ is transformed from the arbitrary b frame into arbitrary a frame via a DCM multiplication.

DCMs have several useful properties [31] that will be utilized in this thesis. Multiple DCM transformations can be chained together through continued DCM multiplications as:

$$\mathbf{C}_a^b = \mathbf{C}_c^b \mathbf{C}_a^c$$

The inverse of a DCM is equal to its transpose which gives the reverse transformation as:

$$\mathbf{C}_a^b = (\mathbf{C}_b^a)^{-1} = (\mathbf{C}_b^a)^T \tag{2.7}$$

Transforming angular rate between reference frames follows the same operation as in (2.6) when the angular rate is expressed in column vector form. Often it is useful to express angular rate in skew symmetric form. Angular rate in skew symmetric form is transformed via a modified DCM equation as:

$$\mathbf{\Omega}_{ba}^b = \mathbf{C}_c^b \mathbf{\Omega}_{ba}^c \mathbf{C}_b^c \tag{2.8}$$

Where $\boldsymbol{\Omega}_{ba}^{b}$ is read as the angular rate from b to a expressed in b, and is the skew symmetric form of $\boldsymbol{\omega}_{ba}^{b}$ expressed as:

$$\boldsymbol{\Omega}_{ba}^{b} = \boldsymbol{\omega}_{ba}^{b}\times \tag{2.9}$$

2.6.2 Specific Coordinate Frame Transformations. Several DCMs are used throughout this thesis and are defined here. One such DCM describes the rotation from the NED frame to the ECEF frame given in [4] as:

$$\mathbf{C}_{n}^{e} = \begin{bmatrix} -\sin(L)\cos(\lambda) & -\sin(\lambda) & -\cos(L)\cos(\lambda) \\ -\sin(L)\sin(\lambda) & \cos(\lambda) & -\cos(L)\sin(\lambda) \\ \cos(L) & 0 & -\sin(L) \end{bmatrix} \tag{2.10}$$

where L is geodetic latitude and λ is longitude. This DCM only represents the rotation required to express a vector previously defined in the NED frame into the ECEF frame. If the geodetic position that was described in the geodetic frame is desired to be expressed in the ECEF frame, a direct conversion is given by [8] as:

$$p_x^e = (R_p + h)\cos L\cos\lambda \tag{2.11}$$

$$p_y^e = (R_p + h)\cos L\sin\lambda \tag{2.12}$$

$$p_z^e = (R_p(1-e^2) + h)\sin L \tag{2.13}$$

Often in this thesis it will we useful to calculate the connecting vector between two bodies whose positions are defined in the geodetic frame. A useful relationship that expresses this quantity is given as:

$$\mathbf{r}^n = \begin{bmatrix} (R_p + Alt_2)(L_2 - L_1) \\ (R_m + Alt_2)\cos L(\lambda_2 - \lambda_1) \\ Alt_1 - Alt_2 \end{bmatrix} \tag{2.14}$$

where R_m is the radius of curvature in the meridian defined in (2.2) and $\mathbf{r}^n$ is the vector that points from a body located at (L_1, λ_1, Alt_1) to the body located at (L_2, λ_2, Alt_2) expressed in the NED frame.

One essential DCM describes the transformation between the NED and the body frame. In terms of the quaternion elements which is the chosen method for attitude representation in this research, the body to NED DCM is given by [29] as:

$$\mathbf{C}_b^n = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{2.15}$$

Because the NED and body frame are colocated on the vehicle, (2.15) describes the complete transformation between the two frames rather than only the rotation as was seen between the NED and ECEF frames. The NED frame is related to the geodetic frame by unit conversions of the north and east elements and the simple rotation:

$$\mathbf{C}_n^g = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.16}$$

where $\mathbf{C}_n^g$ is the rotation from the NED frame to the geodetic frame.

For some sensor measurements, the orientation of the sensor is important. In this thesis, sensor to body frame transformations are handled by a single DCM multiplication to resolve sensor measurements in the body frame:

$$\mathbf{f}^b = \mathbf{C}_s^b \mathbf{f}^s \tag{2.17}$$

where $\mathbf{C}_s^b$ is the sensor to body DCM and $\mathbf{f}^s$ is the measured quantity in the sensor frame. The sensor to body frame DCM includes all information required to resolve the measurement into the body frame, including the sensor frame and the sensor mounting orientation.

2.6.3 Attitude Representation Conversions. Conversions between all three types of attitude representations are necessary for using the DCM for the transformation between body and NED frame and also to transform between measurements of certain sensors. The Euler angles relate to the unit quaternion as:

$$\begin{aligned} q_1 &= \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_2 &= \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_3 &= \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_4 &= \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{aligned} \tag{2.18}$$

The inverse relationship of relating the quaternion parameters to Euler angles has many applications and is used throughout this document primarily to branch certain sensor measurements to estimated states and to give an attitude output that has a meaningful physical interpretation. The conversion is best carried out by forming the body to NED DCM and using the elements to form the relationship as:

$$\begin{aligned} \phi &= \tan^{-1}\frac{c_{32}}{c_{33}} \\ \theta &= \sin^{-1} -c_{31} \\ \psi &= \tan^{-1}\frac{c_{21}}{c_{11}} \end{aligned} \tag{2.19}$$

where the elements of the DCM are given here for clarity:

$$\mathbf{C}_b^n = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{2.20}$$

2.6.4 Quaternion Rotations. Occasionally in this document it will be useful to perform rotations using only the quaternions. The current quaternion attitude can be represented as multiplications of the individual attitude changes. For example, if

a rotation $\mathbf{p}$ has taken place from the inital attitude $\mathbf{q}$, the current attitude $\mathbf{q}_{final}$ is represented by the quaternion multiplication:

$$\mathbf{q}_{final} = \mathbf{Q}(\mathbf{q})\mathbf{p} \tag{2.21}$$

where the operator $\mathbf{Q}$ given in terms of the elements of $\mathbf{q}$ is:

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \tag{2.22}$$

An individual rotation can be extracted from current attitude using the inverse relationship of (2.21). In terms of the previous example, this is expressed as:

$$\mathbf{q} = \mathbf{Q}(\mathbf{q}_{final})\mathbf{p}^* \tag{2.23}$$

where $\mathbf{p}^*$ is the quaternion conjugate derived from its complex definition and is given as:

$$\mathbf{p}^* = \begin{bmatrix} p_1 & -p_2 & -p_3 & -p_4 \end{bmatrix}^T \tag{2.24}$$

2.7 Gravity Models

The acceleration due by gravity is given by the combined effects of the centripetal force from the earth's rotation and earth's gravitational field [4]. Estimating acceleration due to gravity plays a huge role in any navigation that incorporates accelerators not limited to the horizontal directions only. Even slight errors in the gravity estimate will cause large errors in the navigation solution over time.

Multiple methods are employed to estimate gravity such as zero velocity updates and gravity models. Zero velocity updates allow the gravity force magnitude to be estimated if it is known that the platform is stable. The gravity force vector acting

on the sensor is then determined if the attitude is known. Gravity models calculate gravity given a latitude and height above the reference ellipsoid.

For simplicity, this research does not include zero velocity updates. The acceleration due by gravity is then independently determined using a gravity model. The gravity acceleration effects are also mitigated through some aiding altitude, which in this research is provided by any sensor with altitude measurement capabilities. The aiding comes through the routine process of measurements updates in the filter.

One main gravity model considered in this document is a basic model from [29] given by:

$$\mathbf{g}^n = \mathbf{g} - \frac{\omega_{ie_z}^2 (R_0 + h)}{2} \begin{bmatrix} \sin 2L \\ 0 \\ 1 + \cos 2L \end{bmatrix} \tag{2.25}$$

$$\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ g(h) \end{bmatrix}, g(h) = \frac{g(0)}{(1 + h/R_0)^2} \tag{2.26}$$

$$g(0) = 9.780318(1 + 5.3024 \times 10^{-3} \sin^2 L - 5.9 \times 10^{-6} \sin^2 2L) m/s^2 \tag{2.27}$$

where R_0 is the geometric mean of radius of curvature in the prime vertical and in the meridan:

$$R_0 = \sqrt{R_p R_m} \tag{2.28}$$

2.8 Sensor Error Types

Due to the large number of sensors to be described in this document, the characteristics of the errors seen in the associated measurement equations are generically explained in this section. There are two types of errors that are used to create the sensor error models:

- Constant bias

- Time correlated bias (TCB)

The constant bias is straightforward and is generally used to represent scale factor errors and other turn-on turn-off biases. These are error terms that are assumed to change very little throughout a single run. Its propagation is modeled as zero with a very small amount of added white Gaussian noise (WGN) [20]:

$$\dot{b} = 0 + w_b \tag{2.29}$$

The noise strength Q_b of the constant bias is given by the autocorrelation of the WGN as:

$$E[w_b(t)w_b(t+\tau)] = \sigma_b^2\delta(\tau) \tag{2.30}$$

$$Q_b = \sigma_b^2 \tag{2.31}$$

The constant bias propagation is equivalent to Brownian motion.

The TCB is characterized by a time constant T and the WGN term w_b. The propagation of the TCB bias is defined as:

$$\dot{b} = -\frac{1}{T}b + w_b \tag{2.32}$$

The variance and autocorrelation of the TCB bias is given respectively as:

$$E[b(t)^2] = \sigma_b^2 \tag{2.33}$$

$$E[b(t)b(t+\tau)] = \sigma^2 e^{-|\tau|/T} \tag{2.34}$$

Since the steady state value of the autocorrelation of the TCB is the variance σ_b^2 as seen in (2.34) [20], the desired value of the noise strength is:

$$Q_b = 2\frac{\sigma^2}{T} \tag{2.35}$$

2.9 Kalman Filter Equations

Kalman filtering describes a wide field of stochastic state estimation techniques given a system model and state measurement updates. There are many different types of Kalman filters used in practice, but all are based on the conventional Kalman filter. The conventional Kalman filter is an optimal state estimation algorithm as long as certain assumptions are met, such as that system be linear in nature and that state and measurement errors be modeled using zero mean WGN.

In many real world applications, the linearity assumption is not valid, which led to the development of the EKF which will be discussed in Section 2.9.2. The noise assumption is not always valid on real systems, but characterizes many errors on measured quantities seen in the physical world. Often, noise-corrupted entities in hardware are summed together, which by the central limit theorem have probability density functions (PDF) that can be approximated as Gaussian.

In continuous time, a stochastic model is expressed in the state-space format in terms of only first-order differential equations:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.36}$$

where $\mathbf{x}$ represents the system states, $\mathbf{F}(t)$ is the state dynamics matrix, $\mathbf{B}(t)$ is the input matrix, $\mathbf{u}(t)$ represents the vector of deterministic inputs, $\mathbf{G}(t)$ is the noise distribution matrix, and $\mathbf{w}(t)$ represents WGN state uncertainty with noise strength $\mathbf{Q}(t)$. Since the state estimate noise characteristics are described by a Gaussian PDF, the state estimate is fully characterized by its mean and covariance:

$$\hat{\mathbf{x}} = E[(\mathbf{x}(t))(\mathbf{x}(t))^T] \tag{2.37}$$

$$\mathbf{P}_{xx} = E[(\mathbf{x}(t))(\mathbf{x}(t))^T] - E^2[\mathbf{x}(t)] \tag{2.38}$$

The discrete measurement is then represented by:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.39}$$

$$E[(\mathbf{v}(t_i))(\mathbf{v}(t_i))^T] = \begin{cases} \mathbf{R}(t_i) & i = j \\ \mathbf{0} & i \neq j \end{cases} \tag{2.40}$$

where $\mathbf{z}(t_i)$ is the measurement vector, $\mathbf{H}$ is the measurement observability matrix, and $\mathbf{v}(t_i)$ is the discretized measurement noise vector. The measurement is also described by a Gaussian PDF and fully characterized by its mean and covariance.

Next the discrete stochastic model used by the discrete conventional Kalman filter is developed. For simplicity, the t_i and t_j time indexing notation is replaced with the k subscript to denote the discrete time model. The current time is referred to by subscript k and the previous time step is denoted by subscript $k-1$. The discrete state-space model is represented by:

$$\mathbf{x}_k = \mathbf{\Phi}_k\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \tag{2.41}$$

where $\mathbf{x}_k$ are the discrete states, $\mathbf{\Phi}_k$ is the state transition matrix that transforms the states from time t_{k-1} to t_k, $\mathbf{B}_{k-1}$ is the discrete input matrix, $\mathbf{u}_{k-1}$ is the vector of discrete inputs, $\mathbf{G}_{k-1}$ is the discrete noise distribution matrix, $\mathbf{w}_{k-1}$ is the discrete noise vector with discretized noise strength $\mathbf{Q}_k$. In this thesis, no inputs are given to the filter, so from this point on, $\mathbf{B}_{k-1}$ and $\mathbf{u}_{k-1}$ are set to zero and ignored in the Kalman filter equations.

The discrete time measurements take on the form:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.42}$$

$$\mathbf{R}_k = E[(\mathbf{v}_k)(\mathbf{v}_k)^T] \tag{2.43}$$

where each element of (2.42) is the the same as the corresponding element of (2.39), only expressed with different syntax for consistency with its associated model.

To make the transformation from (2.36) to (2.41) an equivalent $\mathbf{\Phi}_k$ and $\mathbf{Q}_k$ must be calculated. These conversions are performed using the matrix exponential operator, $e^{\mathbf{F}\Delta t}$, and the three step Van Loan calculation [30] described below.

The discrete state transition matrix can be formed using the Taylor series expansion of state dynamics matrix. This is expressed to the nth order as:

$$e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \frac{(\mathbf{F}\Delta t)^2}{2!} + ... + \frac{(\mathbf{F}\Delta t)^n}{n!} \tag{2.44}$$

and from [5], the state transition matrix is defined as:

$$\mathbf{\Phi}_k \triangleq e^{\mathbf{F}\Delta t} \tag{2.45}$$

In cases where the step time is very small and the state dynamics are relatively slow, a first order approximation will suffice, giving a simplified equation for the state transition matrix as:

$$\mathbf{\Phi}_k \approx \mathbf{I} + \mathbf{F}\Delta t \tag{2.46}$$

The derivation for the discrete noise power matrix is slightly more complicated than the derivation for the state transition matrix. It involves using the three step van Loan calculation, given as:

$$\mathbf{A}^* = \begin{bmatrix} -\mathbf{F} & \mathbf{G}\mathbf{Q}\mathbf{G}^T \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix} \Delta t \tag{2.47}$$

$$\mathbf{B}^* = e^{\mathbf{A}^*} = \begin{bmatrix} \cdots & \mathbf{\Phi}^{-1}\mathbf{Q}_k \\ 0 & \mathbf{\Phi}^T \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \tag{2.48}$$

$$\mathbf{Q}_k \triangleq \mathbf{B}_{22}^T \mathbf{B}_{12} \tag{2.49}$$

This process is computationally expensive, especially when performed hundreds of times per second. Therefore a first order approximation for $\mathbf{Q}_k$ is very beneficial as well. By using the first order definition of $e^{\mathbf{A}^*}$ and performing an element by element

comparison with (2.47), it is noted that $\mathbf{Q}_k$ can be written in terms of $\mathbf{\Phi}$ and the continuous time noise distribution quantity $\mathbf{GQG}^T$ as:

$$\mathbf{Q}_k = \mathbf{\Phi GQG}^T \Delta t \tag{2.50}$$

Using the first order approximation of $\mathbf{\Phi}$ and ignoring higher order terms, this then gives a first order approximation of $\mathbf{Q}_k$ as:

$$\begin{aligned} \mathbf{Q}_k &\approx [\mathbf{I} + \mathbf{F}\Delta t]\mathbf{GQG}^T \Delta t & (2.51) \\ &\approx \mathbf{GQG}^T \Delta t & (2.52) \end{aligned}$$

2.9.1 Kalman Filter Equations. Given a discrete stochastic system model and initial conditions, the Kalman filter will provide state estimates over time by propagating the states as:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1} \tag{2.53}$$

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1} \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{2.54}$$

In the absence of measurements, the state continues to be propagated and the state covariance steadily grows. Once a measurement is available to the sensor, the Kalman gain $\mathbf{K}_k$ is computed as:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k]^{-1} \tag{2.55}$$

The estimate is then updated with the measurement, $\mathbf{z}_k$, and the best state estimate and its associated covariance is them computed as:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) & (2.56) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \end{aligned}$$

The measurement residual and corresponding covariance are given by:

$$\begin{aligned} residual &= \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^- && (2.57) \\ E[(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k)(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k)^T] &= \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}_k && (2.58) \end{aligned}$$

where it is assumed that state estimate and the measurement covariance are independent. Next, the filter propagates forward in time ahead using the dynamics model and current best estimate until the next measurement is available at which point the measurement update equations are applied again.

As long as the Kalman filter assumptions are correct, the residuals will be zero mean WGN. The residuals are like the heartbeat of the Kalman filter in that monitoring them tells how well the system is doing. Residuals exceeding the residual uncertainty level by more than three standard deviations indicate that either the measurement is corrupted, or the system is not being correctly modeled or tuned.

2.9.2 Extended Kalman Filter. Most real systems do not have linear dynamics. This violates a fundamental conventional Kalman filter assumption, eliminating the conventional Kalman filter from use. Hence, the extended Kalman filter (EKF) and other nonlinear dynamics capable Kalman filers were developed. In the EKF, it is still assumed that the PDF of the transformed random variables is accurately estimated using zero mean independent WGN sources by linearizing the system dynamics and measurement equations shown in (2.36) and (2.43). The nonlinear system dynamics and/or measurement equations are generally expressed using:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.59}$$

$$\mathbf{z}(t_i) = \mathbf{h}\left[\mathbf{x}(t_i)\right] + \mathbf{v}(t_i) \tag{2.60}$$

These equations are then relinearized after each state estimate propagation. This creates a new, updated state trajectory every time step. This process minimizes

the difference between the nominal and true trajectories, improving the overall model [19].

The first order linearization is performed by forming the Jacobian of the state dynamics equations evaluated with the current best state estimate given from the last update or from the initial conditions if this is the starting iteration.

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-} \tag{2.61}$$

The state estimate is propagated to the current time step by solving the non-linear differential dynamics equations at the current time with the initial conditions of the current best state estimate. Solving these differential equations hundreds of times per second is computationally expensive. This integration can be simplified by using the state transition matrix to transition the state to the next time step as is the case with the conventional Kalman filter as:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1} \tag{2.62}$$

where the state transition matrix and discrete noise power matrix are calculated using the same equations used in the conventional Kalman filter for the state transition matrix (2.46) and the discrete noise strength matrix (2.51), except the state transition matrix is now calculated using (2.61). The propagation using (2.62) is only valid when the time step is relatively small and the dynamics are relatively slow compared to update times. Fortunately, the simplified forms for the state transition matrix, noise strength matrix, and nonlinear propagations are most needed when the step times are small, as many propagations must be computed. For both the EKF and conventional Kalman filter, the state estimate covariance is propagated to the next time step using:

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1} \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{2.63}$$

Once a measurement is presented to the filter, the measurement equation relating the estimated states to the measurement entity, is linearized about the propagated state. The Jacobian of the measurement observation equation is formed as:

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-} \tag{2.64}$$

Because linearization process, the actual measurement presented to the filter is the measurement residual. With some modification to (2.56), the EKF measurement update equation is then expressed as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) \tag{2.65}$$

where

$$\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}^-) \tag{2.66}$$

The Kalman gain and the estimated covariance are found using the same calculation as in the conventional Kalman filter repeated here for consistency:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k]^{-1} \tag{2.67}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{2.68}$$

Like the conventional filter, these propagation and update steps are iterated throughout the execution of the EKF.

2.9.3 Delayed-State Kalman Filter Modification. Delayed-state sensors present a unique challenge to the navigation solution. Rather than presenting a direct measurement of a state, a delayed-state sensor provides the change of a state from some previous time. The problem with this measurement is that it provides information about the state over the entire time period rather than at the instant the

measurement is obtained. Thus the measurement is modeled rather as:

$$\mathbf{z} = \int_{t_{k-1}}^{t_k} \mathbf{x}\,\mathrm{d}t \tag{2.69}$$

Considering a specific example where the measurement is a position change, the measurement update can be expressed as the integrated velcoity of the valid time interval as:

$$\begin{aligned} \mathbf{z} &= \int_{t_{k-1}}^{t_k} \mathbf{v}\,\mathrm{d}t \\ &= \text{position at } t_k - \text{position at } t_{k-1} \end{aligned} \tag{2.70}$$

The general relationship can be expressed in conventional Kalman filter notation as:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k^- + \mathbf{J}_k \mathbf{x}_{k-1} + \mathbf{v}_k \tag{2.71}$$

In the case of the EKF, the measurement update is modeled as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k^-, \mathbf{x}_{k-1}) + \mathbf{v}_k \tag{2.72}$$

with $\mathbf{h}_k$ represents the nonlinear state measurement equations, and current state and $\mathbf{H}_k$ and $\mathbf{J}_k$ are given by:

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x_k}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \hat{\mathbf{x}}_{k-1}}, \ \mathbf{J}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x_{k-1}}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \hat{\mathbf{x}}_{k-1}} \tag{2.73}$$

The filter delayed-state update equation is then expressed for the EKF as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{2.74}$$

$$\hat{\mathbf{z}}_k^- = \mathbf{h}(\mathbf{x}_k^-, \mathbf{x}_{k-1}) \tag{2.75}$$

where the Kalman gain is derived in [5] as:

$$\mathbf{K}_k = [\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{J}_k^T]\mathbf{L}_k^{-1} \tag{2.76}$$

where $\mathbf{L}_k$ is the residual covariance given by:

$$\mathbf{L}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k + \mathbf{J}_k\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^T\mathbf{H}_k^T + \mathbf{H}_k\mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{J}_k^T + \mathbf{J}_k\mathbf{P}_{k-1}\mathbf{J}_k^T \tag{2.77}$$

The state covariance update is given by:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{L}_k\mathbf{K}_k \tag{2.78}$$

The final measurement update is unchanged and performed using the current best state estimate. The state propagation equations for the delayed-state are the same as with the conventional Kalman filter for the state estimate (2.53) and the state covariance (2.54). For a thorough derivation of the delayed-state Kalman filter equations, the reader is referred to [5].

2.9.4 UD Factorization. In real world implementations of the Kalman filter, numerical precision can sometimes become an issue. Thanks to modern computing, this does not occur very often. One situation that could lead to numerical issues would be when updating a large covariance matrix with a very accurate measurement. The upper triangular/diagonal(UD) factorization Kalman filter propagate and update equation modifications seek to minimize error caused by limited computer numerical precision. Then general idea of the UD factorization is to reduce the wide dynamical range of the state covariance matrix during propagation and update operations [5]. By performing the UD factorization, the filter ensures that the iterated covariance matrix $\mathbf{P}_k$ remains positive semi-definite. UD-factorization comes at a price, however, as it requires more math operations performed for every filter iteration, considerably slowing down the filter.

UD factorization is based on the fact that a symmetric, positive matrix $\mathbf{P}$ can always be decomposed into the UD factored form:

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{2.79}$$

where $\mathbf{U}$ is the upper diagonal matrix with a diagonal of ones, nontrivial elements above the diagonal and zeros below, and $\mathbf{D}$ is diagonal [5]. This factorization is especially useful in this research were many hundreds of updates will be made per second and numerical stability could be a concern. The process for implementing the UD factorization is well described in [5, 20] and the reader is encouraged to refer to these resources for the specific steps and equations. This thesis implements UD Kalman filter equations for both the covariance update and covariance propagation.

2.10 Unscented Transform

The unscented transform (UT) provides a valuable tool for estimating the mean covariance of Gaussian random variables transformed by nonlinear equations. The basic operation involves running strategically placed sigma points through the nonlinear equations and then estimating the mean and covariance of the resulting distribution of sigma points. The UT was used in this research primarily in post or preprocessing scenarios such as converting initial values of roll, pitch, and yaw into quaternion elements without computing the need to compute complicated Jacobian matrices. The reader is referred to the original paper by Julier and Uhlmann [14] where the UT is described in detail. The description presented here is the basic idea as required for this research.

In the unscented transform, $2N + 1$ sigma points are generated where N is the number of input parameters. The sigma points are generated using:

$$\boldsymbol{\chi} = [\hat{\mathbf{x}}\ \hat{\mathbf{x}} \pm \sqrt{N + \Upsilon}\sqrt[c]{\mathbf{P}_{xx}}] \tag{2.80}$$

Where $\boldsymbol{\chi}$ is a $N \times 2N + 1$ matrix and Υ is a weighting factor given by:

$$\Upsilon = \alpha^2(N + \kappa) - N \tag{2.81}$$

with tuning parameters α and κ set to $\alpha = 0.001$ and $\kappa = 0$. The sigma points are then passed through the dynamics equation:

$$\boldsymbol{\Psi} = \mathbf{f}(\boldsymbol{\chi}) \tag{2.82}$$

The resulting mean and covariance of the of the sigma points are calculated using:

$$\begin{aligned} \hat{\mathbf{x}} &= \sum_{i=0}^{2L+1} \mathbf{W}_i^m \boldsymbol{\Psi}_i \\ \mathbf{P}_{xx} &= \sum_{i=0}^{2L+1} \mathbf{W}_i^c (\boldsymbol{\Psi}_i - \hat{\mathbf{x}})(\boldsymbol{\Psi}_i - \hat{\mathbf{x}})^T) \end{aligned} \tag{2.83}$$

where weights $\mathbf{W}$ are given as

$$\begin{aligned} \mathbf{W}_0^m &= \Upsilon/(N + \Upsilon) \\ \mathbf{W}_0^c &= \Upsilon/(N + \Upsilon) + 1 - \alpha^2 + \beta \\ \mathbf{W}_i^m = \mathbf{W}_i^c &= 1/[2(N + \Upsilon)] \; i = 1, ..., 2N \end{aligned} \tag{2.84}$$

and β is another tuning parameter that is set to $\beta = 2$ for Gaussian.

2.11 Previous Research

Many of the sensors used in this thesis have been integrated together many times to provide synergistic effects. It is common to find GNSS sensors fused with an IMU or cameras fused with an IMU in literature [10, 15, 16, 31]. This research separates itself from other efforts by the quantity and variety of the sensors combined together. No other open research project rivals this thesis in the variety of sensor data. Because of this, this section will primarily cover smaller research projects dealing with clusters

of sensors that are of interest in this project. These projects aided and provided insight into the process of building the filter and supporting code for this thesis.

2.11.1 Bancroft/Lachapelle Data Fusion Algorithms for Multiple Inertial Measurement Units. Bancroft and Lachapelle considered advances in micro electro-mechanical systems (MEMS) allowing access to cheap and physically reliable IMU systems. This is of great potential interest to consumers such as the military, search and rescue, athletes, and the visually impaired to name a few. Bancroft and Lachapelle discuss virtual IMU (VIMU) observation fusion, centralized filter design, and federated filter design implementations. Past implementations are described including the prominent method of redundant IMU integration using LMS in which errors are not modeled or fault tested prior to fusion which is fundamentally flawed [1].

The VIMU LMS method implements a nine parameter estimation model that includes angular rate, angular acceleration, and the specific force vectors. This requires the use of at least three IMUs to obtain a full rank design matrix. The VIMU filter requires all IMU updates to be time synchronized. The VIMU does not estimate the IMU errors prior to fusion but estimates the the error in the combined solution. This limits the ability to perform fault detection on the VIMU and requires artificial increases in the covariance matrix to account for the unmodeled errors.

The centralized filter is composed of several separate filter blocks. Each IMU has its own filter block which contains 21 states that represent the position, velocity, attitude, biases and scale factors. Each block is updated individually with a time synchronized prediction cycle. The main advantage, other than simplifying time synchronization, is that the IMU errors are able to be estimated. The main disadvantage is that blocks are updated by a GPS which must be used repeatedly for each IMU.

The federated filter also contains miniature 21 state filter blocks for each IMU. The position, velocity, and attitude states are shared, while the IMU error states remain unchanged in the filter blocks. Two separate types of federated filters, the federated no reset and the federated fusion reset are discussed.

The methods of redundant IMU integration discussed in [1] provide a well described overview of multi-IMU integration. In particular, the centralized filter presented the most straightforward approach and most often the best results. However the IMU setups in [1] were rigidly defined, at least in comparison to the requirements of this thesis. Overall the main concept that stands out, is that multi-IMU integration is a challenging task and only recently with the advent of cheap effective systems has it come under close review.

2.11.2 Soloviev/de Haag Scanning Laser Three Dimensional Navigation. Soloviev and de Haag [26], investigated the use of a scanning Laser Radar (LADAR) for three dimensional navigation with an INS in structured environments. The navigation includes using laser scans to provided estimates of both attitude and position changes.

The hardware employed for the scans was a two dimensional LADAR (SICK LMS-200) augmented with a servo motor to enable three dimensional scanning. Multiple lines extracted from scans of planar surfaces are then used to form three dimensional planes which are then assigned a unique feature identification number. The planes must be kept track of with certainty in subsequent frames and this is performed using a feature matching procedure that exploits INS to predict plane location changes. Motions and angular rates that exceed LADAR factory error specifications must also be corrected using an INS.

Plane location change was then related to the body position and described through geometry by the change of the closest point to the observed plane. The position change was then calculated using least mean squares (LMS) and required at least three noncolinear planes. The position change accuracy is borrowed from the GPS dilution of precision (DOP) calculation and assumes all the components of the plane normal have equal uncertainty.

Attitude change was given by the rotation of the planar normals between scans. It was determined through a multiple step process that involved estimating the atti-

tude change DCM that translates a specific plane normal from one scan to the next using LMS. The LMS attitude estimation requires at least two noncolinear planes to not be underdetermined.

Soloviev and de Haag provided a useful method for navigating using three dimensional planes including determination of attitude without solving nonlinear equations. The disadvantage to the laser scanner navigation method in [26] is that no useful information about the position or attitude can be obtained when the system is underdetermined. Another disadvantage is that the planes could be better utilized if more was known about the plane parameter covariances. Implementing weighted least mean squares (WLMS) with the associated DOP calculation from [23] could be a possible method to provide greater accuracy.

2.11.3 Schneider/et al. Fusing Vision and Light Detection and Ranging. Schneider/et al. take the a more tightly coupled approach of optical/laser scanner navigation by forming a direct sensor fusion relationship. The procedure invloves fusing monocular vision and three dimensional light detection and ranging (LiDAR) by matching points observed by the scanning LiDAR with visible points in the camera image. This method has recently received much attention and adds a great deal to the navigation field. Coupled together, the camera and LiDAR scanner mitigate disadvantages seen in the single use. While LiDAR scanner has the advantage of detailed three dimensional scans, it has inferior angular resolution and cannot provide color which can be useful for object detection [24]. The moncular camera on the other hand provides color and superior angular resolution when calibrated correctly, but is unable to provide depth.

LiDAR and vision fusion poses several problems for implementation that are addressed and are a main contribution of [24]. These difficulties include sensor time synchronization and occlusion. Time synchronization is handled by shifting the camera exposure time so the exposure is taken as the beam passes the field of view of the camera. The LiDAR scanner continuously takes data so its time synchronization is

handled by the bearing information from the incoming data packets. Occlusion arises from different physical locations of the sensors causing separate scenes to be viewed. This problem is solved using a cloud segmentation algorithm.

The camera/LiDAR fusion in [24] provides numerous possibilities for navigation. Pixels with depth information provided by the sensor set could potentially be matched with camera extracted features allowing for accurate feature initialization, especially with monocular cameras. Color matched with depth information could also be used for a very reliable object identification scheme, where cataloging might be employed with stored object locations. The main disadvantages seen in [24] are the additional computational burden of solving the occlusion problem, extra hardware configurations, and sensor synchronization. Ultimately the combination provides an excellent area for an additional type of sensor. The LiDAR scanner could still be used separately, since [24] only uses a fraction of the information available from the LiDAR scanner.

2.12 Summary

In this chapter, the math notation used throughout this document was discussed. Reference frames are presented as well as the preferred method of rotating between frames using the DCM. Three main attitude representations advantages and disadvantages are discussed as well as their relationships to each other. The attitude representation of choice is chosen from the three main attitude types to be the unit quaternion.

Physical models and quantities required for real world navigation were discussed including the WGS-84 ellipsoid reference model and a gravity model. The conventional Kalman filter was described as well as the EKF to handle nonlinear dynamics. The numerically precise UD Kalman filter update and propagate equations and the delay-state measurement update equations that account for accumulated type measurements were discussed as well. The UT, an alternate method for estimating the the covariance of random variables transformed by nonlinear equations was also described.

Previous research discussed includes projects that combine multiple IMUs, LiDAR and vision fusion and LADAR navigation from three dimensional planes. For each of these research projects, the main interest and method is discussed as well as the potential advantages and disadvantages of the work. Some of these previous works contained concepts that were built off of in this thesis while another project provides future areas of research.

III. Filter Design and Implementation

This chapter outlines the development of the means necessary to combine the observations from the vast sensor set in order to provide an optimal navigation solution. This is a challenging task given the diverse set of requirements for the massive sensor test bed. A basic dynamics model is developed that is tunable according to the dynamics of each platform with minimal complexity. The filter implementation that made it possible to relate the diverse requirements of this thesis to the EKF is described. Sensor background information required to complete this thesis is presented as well as relevant implementation equations.

3.1 Generic Platform Dynamics

Part of the ASPN methodology was to provide a standardized complete set of data to performers, along with a "standard" solution and truth data. In order to provide the standard solution, a very simple generic dynamics model was adopted. Scenarios were designed to test navigation in multiple environments with multiple types and quantities of sensors. Without the rigid setup definition that is available in most projects where an INS is fitted with a GPS or stereo vision with an INS, a very flexible approach had to be taken. Thus came the insight for implementing the EKF which is a very well known and often implemented filter.

All sensor measurements are treated as traditional EKF measurement updates, including INS data. Often in practice, INS measurements are not treated as measurement updates. A traditional use for the INS is providing a reference trajectory, and the Kalman filter estimates the errors in the INS trajectory using some other aiding method such as GPS [5]. This error state approach is advantageous in that it only requires the estimation of position, velocity, and attitude states, where the full-state method additionally requires states for acceleration and angular rate. The main reason the error state approach was not adopted in this thesis was because of the fundamental goal of not having rigidly defined sensor sets. Scenarios with intermittent INS were considered a possibility as well as scenarios with multiple INS. Though

methods of combining multiple INS are available in literature, it is still a fairly recent subject of interest [1], and the better methods are generally not trivial or flexible.

3.2 Dynamics Model Derivation

In this Section, the navigation states are chosen and a general dynamics model is formed. Taking into account treatment of the IMU measurements as measurement updates, the required basic navigation states to be estimated are selected as:

$$\mathbf{x} = \left(\begin{bmatrix} L & \lambda & Alt & (\mathbf{v}^n)^T & (\mathbf{q})^T & (\mathbf{a}^b)^T & (\boldsymbol{\omega}_{nb}^b)^T \end{bmatrix}_{1\times 16} \right)^T \tag{3.1}$$

where L, λ and Alt are the geodetic latitude, longitude and altitude described in Section 2.2, $\mathbf{v}^n$ is the NED velocity, $\mathbf{a}^b$ is the body frame acceleration, $\mathbf{q}$ is quaternion based body attitude, and $\boldsymbol{\omega}_{nb}^b$ is the body to NED frame angular rate expressed in the body frame. All navigation states refer to a centrally chosen location on the platform for the navigation system.

Forming the propagation equations for the states [29], the position states propagate as:

$$\dot{L} = \frac{v_N}{R_m + Alt},\ \dot{\lambda} = \frac{v_E \sec L}{R_p + Alt},\ \dot{Alt} = -v_D \tag{3.2}$$

where v_N, v_E, and v_D are the x, y, and z components of $\mathbf{v}^n$. The velocity and acceleration of the mobile platform, $\mathbf{v}^n$ and $\mathbf{a}^b$, are the velocity and acceleration caused only by the body's motion. Thus the effects of the gravity, Coriolis, motion over a curved surface, etc. are not considered in the dynamics equations, but only in the measurement update equations. This leads to the simple Newtonian definition of the propagation of velocity as the body acceleration rotated into the NED frame as:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{a}^b \tag{3.3}$$

A FOGM acceleration model is adopted to describe the acceleration dynamics as shown in (3.4). This model is tuned to reflect the dynamics of the specific platform

being observed. This is done in a very generic sense with regard to the physical properties, expected performance and maneuvers. In one axis, the acceleration propagation is given as:

$$\dot{a}^b = -\frac{1}{T_{a_x}} a^b + w_{a_x} \tag{3.4}$$

where time constant T_{a_x} and noise parameter w_{a_x} is user defined and can be tuned according to platform dynamics. The relationship is the same for the other two axes.

From [29], the quaternion attitude propagates as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = 1/2\mathbf{Q}(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \tag{3.5}$$

where the definition of $\mathbf{Q}$ is given in (2.22) and repeated here for convenience:

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \tag{3.6}$$

Using a FOGM angular rate model tuned to the specific platform, the NED to body frame angular rate in one axis propagates as:

$$\dot{\omega}_{nb_x}^b = -\frac{1}{T_{\omega_x}} \omega_{nb_x}^b + w_{\omega_x} \tag{3.7}$$

where the time constant T_{ω_x} and noise parameter w_{ω_x} is user defined and can be tuned according to platform dynamics. The relationship is the same for the other two axes as well. From the vector of combined propagation equations $\mathbf{f}$, the model takes on

the form seen in 2.59 and repeated here for clarity.

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \mathbf{G}(t)\mathbf{w}(t) \tag{3.8}$$

Where $\mathbf{G}(t)$ is the noise distribution matrix which for the basic dynamics model takes the form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \vdots & \vdots \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}_{16\times 6} \quad \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_\omega \end{bmatrix}_{6\times 1} \tag{3.9}$$

with associated noise power matrix $\mathbf{Q}$ given by:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_a & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{Q}_\omega \end{bmatrix} \tag{3.10}$$

Where $\mathbf{Q}_a$ and $\mathbf{Q}_\omega$ are derived from the time constants and noise variances of the FOGM acceleration and angular rate on each the three axes. This is described for one axis for both state quantities as:

$$Q_a = \frac{2\sigma_{a_x}^2}{T_{a_x}},\ Q_\omega = \frac{2\sigma_{\omega_x}^2}{T_{\omega_x}} \tag{3.11}$$

3.3 State Space Model

To place the dynamics equations into state space form as seen in equation 2.36, the Jacobian matrix is formed from the partial derivatives of the nonlinear equations as:

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.12}$$

No inputs are considered in this thesis so the input distribution matrix $\mathbf{B}$ is set to zero. Ignoring second order terms, the derived basic linear dynamics matrix $\mathbf{F}$ is given in (3.13). This basic model form will be used in all platform runs regardless of the sensor set used.

$$\mathbf{F} = \left[\begin{array}{ccc|ccc|cccc|cccc|ccc} 0 & 0 & 0 & \frac{1}{R_m+Alt} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\cos L(R_p+Alt)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{21} & c_{22} & c_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{31} & c_{32} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{w_x}{2} & -\frac{w_y}{2} & -\frac{w_z}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{w_x}{2} & 0 & \frac{w_z}{2} & -\frac{w_y}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{w_y}{2} & -\frac{w_z}{2} & 0 & \frac{w_x}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{w_z}{2} & \frac{w_y}{2} & -\frac{w_x}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{a_x}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{a_y}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{a_z}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\omega_x}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\omega_y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\omega_z}} \end{array}\right] \tag{3.13}$$

where ω_x, ω_y, ω_z are the elements of the body to NED frame angular rate as:

$$\boldsymbol{\omega}_{nb}^{b} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \tag{3.14}$$

and the components of the body to NED frame direction cosine matrix (DCM) are given as:

$$\mathbf{C}_{b}^{n} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{3.15}$$

The basic navigation states are relisted here for completeness:

$$\mathbf{x} = \left(\begin{bmatrix} L & \lambda & Alt & (\mathbf{v}^n)^T & (\mathbf{q})^T & (\mathbf{a}^b)^T & (\boldsymbol{\omega}_{nb}^{b})^T \end{bmatrix}_{1\times 16} \right)^T \tag{3.16}$$

3.4 Filter Design

The filtering code was required to handle multiple sensors types each possessing its own specific set of requirements. Some sensors required the addition of error states at the sensor introduction while others required dynamic sensor monitoring and state addition and removal. Many sensors required no additional states, but provided an unknown number of measurements per epoch.

The key to successful filter implementation was ensuring that the filter was flexible enough to accommodate all sensors requirements, while being uniform enough not to require numerous modifications per additional sensor incorporation. The coding was completely handled in `Matlab`® , and is described by the top level flow chart given in Figure 3.1. The following sections provide descriptions of the illustrated routines.

3.4.1 Load Settings and Parameters. This portion of the filter contains all user defined parameters set prior to each filter run and is the first step in the initialization stages. This includes items such as whether real or simulated data will be used, lever arm effect estimation and error state parameters for sensors that do

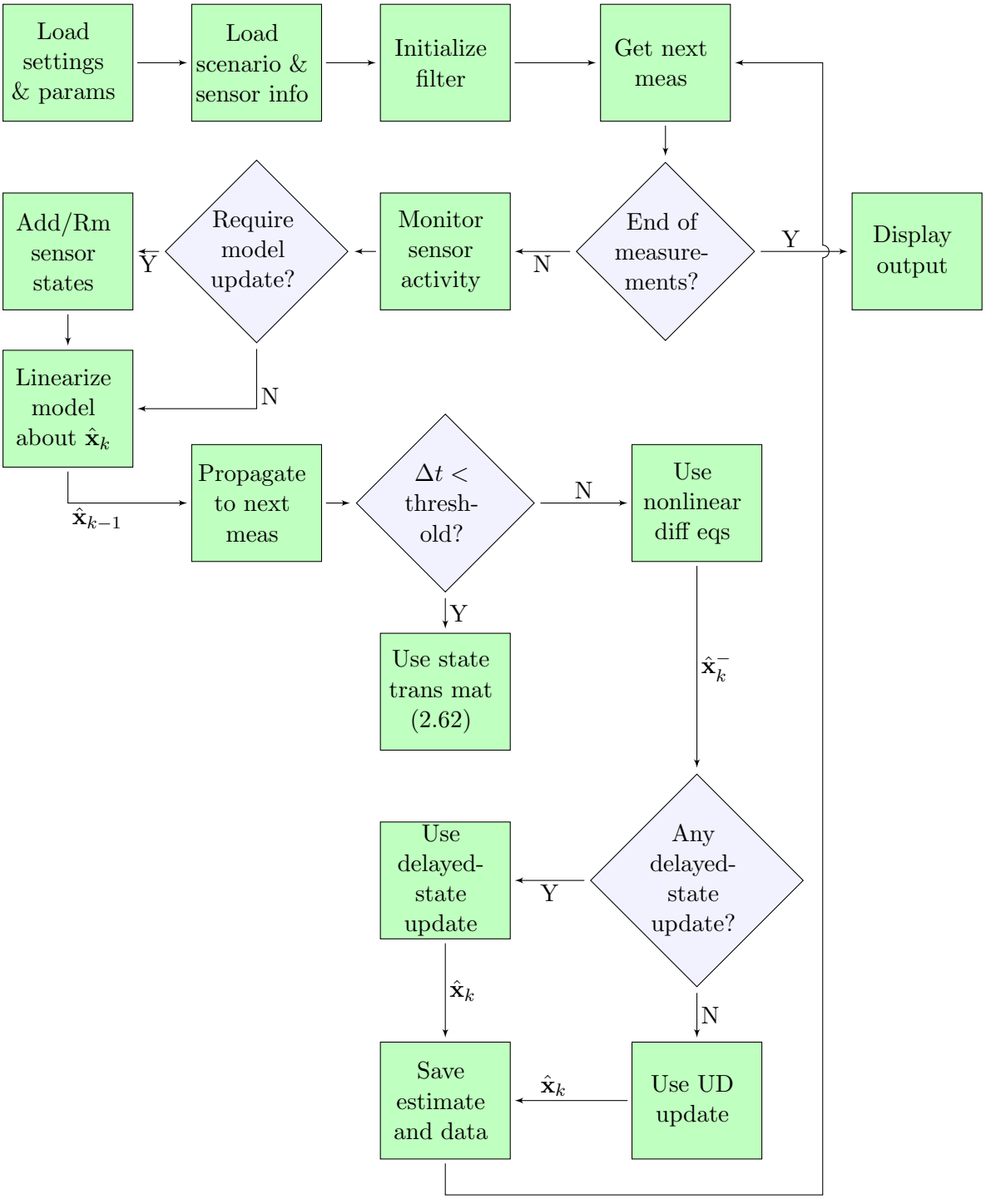

Figure 3.1: Top level filter flow diagram

not provide it (Note: only simulated data and results are used in this thesis due to the lack of availability of ASPN real data at the time of writing). After this process, a configuration structure is created.

3.4.2 Load Scenario and Sensor Information. Once the user defined commands are stored, the filter reads in the scenario and sensor information. Each scenario is described using a configuration file. The configuration file provides essential information about the setup including platform type, initial state values, sensor lever arms, orientation, sensor names, and sensor file names. The sensor lever arms are defined in the body frame. The orientation of the sensor includes both information with regard to its orientation on the platform and its specific sensor frame. That is, the orientation DCM rotates quantities directly from the sensor frame into the body frame. The sensor file names refers to files that contain sensor metadata, file information, and measurements. The metadata describes the properties of the sensors such as measurement noise and error state characteristics, as well as database information for sensors that require it. File information describes items such as the number of measurements taken, and sensor start and stop times for coding only.

From this step, the filter is aware of every sensor that will appear at some point during the run. At no point in the filter is this information used to improve the estimate. It is only used to provide simpler code layout and lower computational burden where possible. Most sensor parameters such as measurement noise strength and error state values are contained in the sensor metadata.

Once all the available sensor information is retrieved and stored, the filter is initialized. Initial state estimates are provided in terms of the mean and covariance. All provided initial state means are provided in the same units as the estimated states, with the exception of the attitude which is provided as a DCM. The initial attitude is converted from DCM to quaternion representation using the UT of (2.19) substituted into (2.18) which provides an estimated quaternion mean and covariance. The position covariance is provided in units of meters which is readily converted to

the estimated states covariance units of rad^2 through (3.17).

$$\mathbf{P}_{\mathbf{p}^n} = \begin{bmatrix} \frac{1}{R_E^2} & 0 & 0 \\ 0 & \frac{1}{(R_E \cos L)^2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{P}_{\mathbf{p}_0^n} \tag{3.17}$$

where R_E is the mean radius of the earth and $\mathbf{P}_{\mathbf{p}_0^n}$ is the initial position covariance. All other covariances are then given in the units of the estimated state.

3.4.3 State Descriptor. Multiple state additions and removals are performed throughout runs depending on the sensors to be implemented; therefore, no set state setup could be assumed at any time during the filter run. A method was devised to track filter states using a state descriptor structure. Sensors are allotted user-defined state descriptors which are unique identification keys. Since the list of sensors that will be used at some point during the run is provided in initialization, each sensor is assigned with a finite number of descriptors. Sensors that have a set number of states are given a fixed number of required descriptors. Sensors that have changing state requirements are given a set amount that are recycled throughout filter operation.

The descriptors are saved in the main state structure as separate fields under the name that the sensor has been given at filter initialization. When states are added, as will be explained later, the descriptors of all estimated states are copied into a single state descriptor vector. At the start of a run, the state descriptor vector contains only the addresses of the basic 16 navigation states, but as sensors are added, the number of held keys increases. The indices of the addresses in the state descriptor vector correspond to the same indices of the states in the state estimate vector and covariance matrix. That is, a particular sensor state can be retrieved from the state vector by copying the sensor's unique descriptor keys stored under its name and searching the state descriptor vector for those keys. When the keys are matched in the state descriptor, the corresponding indices of the match give the indices of the state quantities in the state estimation vector.

Along with keys, idle time limits are assigned to the sensors in order to prevent estimating states that are no longer relevant, such as when a sensor is no longer operating. The idle time is calculated as the current time minus the time the sensor last provided a measurement. When a sensor's idle time exceeds a predetermined threshold, the sensor states are removed from the filter set of estimated states.

3.5 Measurement Retrieval

The measurement retrieval routine's primary concern is obtaining the measurement from a buffered data structure. The data structure is populated by the binary data files that contain the sensor measurements. Buffering is only performed here to reduce computation time for opening and closing files. The only filter control this function possesses is the ability to end filter iteration when it detects the end of measurements. All other control is providing to the sensor management routine described next.

3.6 Sensor Activity Monitoring

The sensor management routine acts on the data provided after the measurement is retrieved. This includes calculating sensor idle times, updating measurement numbers for variable measurement number sensors, providing the sensor state addition/removal flag, and all other sensor operation information.

3.6.1 Camera Management. Tracking features of multiple cameras requires a significant amount of bookkeeping. Each feature requires the addition of three states to the state vector, potentially adding significant computational burden. Feature ID tracking lists are zero padded to the user defined max feature amount per camera. Since tracking features is computationally expensive and there is no need to track features that are not receiving new measurements, each feature idle time is monitored at every camera measurement update. Every time a camera measurement is available, the measurement feature IDs are checked with the current tracked feature list. Positive

identifications have their idle times set to zeros, are matched, and then saved for measurement update. Tracked features that are idle longer than the user defined threshold are zeroed out in the feature ID tracking list.

New features in the measurement replace zero spots in the tracking list up to the available amount. Left over zero elements of the tracking list that were not previously zero at the completion of the last update are then flagged for deletion. The tracking list is then sorted by moving all zero elements to the end of the tracking list. The total number of tracked features is then given by the number of non-zero elements of the tracking list. The information about the camera is then passed to add and remove state routines as well as the measurement update routine.

3.6.2 Delayed-State Sensor Management. Multiple delayed-state sensors cannot be handled in the measurement update equation as given in Chapter II. During sensor management, the update sensor list is checked for multiple delayed-state sensors. Multiple instances are incorporated serially by allowing a zero time step propagation for as many times as it takes to include all the sensors.

3.6.3 Activity Masks. In addition, this routine provides logical masks describing all sensor activity to remove redundancy. The activity masks refer to lists that contain items such as sensor names, type, measurement number, etc. This is used to find whether sensors are considered operational, names of sensors providing measurements this time step, sensors requiring states, measurement vector size buffering, etc. These masks are reused wherever necessary in the remainder of the filter.

3.7 Addition and Removal of States

Based off information provided by the sensor activity monitoring routine, sensor states are added or removed. Generally this is due to the addition of a new sensor or sensor stagnation, but with the camera is performed constantly as new features are

observed and stagnant features are removed. For non-cameras, the process is routine and straightforward.

3.7.1 State Addition. Given a list of names of additional sensors, the number of additional required states is calculated. Looping through the sensor list, the states are initialized and appended to the end of the state estimate vector. The state covariance is initialized as well and appended to the bottom right corner of the state covariance matrix. Most sensor states are error states and thus initialized to zero with zero cross-covariance to other states. Exceptions to this rule are explained on a case by case basis in Sections 3.11.1 - 3.11.17. Assigned sensor addresses are appended to the state descriptor and provide physical location correspondence between the state estimate vector and covariance matrix for future use.

3.7.2 State Removal. The main difference between state addition and state removal is that no knowledge of the state descriptor is required to add states. State deletion is handled by compiling the list of states to be deleted as the sensor change list is iterated through. The states are not actually deleted until the iterations are complete. Deletion of the state estimate involves deleting the state quantities from the state vector and removing the corresponding covariance terms from the covariance matrix. Removing the covariance is illustrated in one dimension by locating the state and then removing the entire row and column that it touches.

3.7.3 Camera State Addition/Removal. Camera states require special treatment, since their addition and removal occur on such a regular basis, and deletion of a state does not mean the removal of the sensor. Rather than only deleting states, the camera replaces them with new available features. This comes directly from the fact that cameras provide so many features that it is impractical to add states for every feature. As such, many features must be ignored until room is available. For compatibility with the other sensors, an additional routine was added so that replacements are handled separately from the add and remove sensor routine. Replaced states are

simply cleared of the past estimated values and re-initialized in value and covariance using the procedure described later in Section 3.11.3. The state descriptor remains the same. States that must be deleted are reported to the main addition and removal state routine and treated like normal state removals.

3.7.4 State Addition/Removal Residual Clean Up. Delayed states require the state transition matrix that was valid at the previous measurement to be resized for each delayed-state sensor when the state estimate vector size changes. This is accomplished for added states by zero padding and appending the identity matrix with size equal to the number of additional states. For state deletions, the elements of the state transition matrix are deleted in the same manner as the covariance matrix. This method is valid, because sensor states are not dynamically coupled with any other states. This step ensures matrix dimensions agree for the delayed-state update.

3.8 Model Linearization

Model linearization occurs every time step about the previous measurement update. Indexing masks, provided from matching sensor state descriptor keys in the state descriptor vector, describe the vertical and horizontal placement of the noise in the noise distribution matrix $\mathbf{G}$ which is populated as:

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{I} & & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}_{N \times W} \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_\omega \\ \vdots \\ \mathbf{w}_N \end{bmatrix}_{W \times 1} \tag{3.18}$$

where W is the total number of required noise terms and N is the total number of states. The associated basic model noise strength matrix $\mathbf{Q}$ is given by:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_a & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_\omega & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{Q}_N \end{bmatrix}_{W \times W} \tag{3.19}$$

3.9 State Propagation

State propagation is handle differently, depending on the step size between propagations. The dynamics matrix $\mathbf{F}$ generally changes slowly compared to the frequency of measurements received, especially since high rate IMU measurements are almost always available. Given a user-defined max time step threshold based on the current platform, the state propagation for very small time steps is handled by a first order linear propagation (2.62). For large time propagations, the nonlinear dynamics equations must be integrated. This thesis uses the `Matlab`® *ODE45* algorithm which employs a Runge Kutta integration scheme for the nonlinear equation integration. The state covariance is propagated using the UD propagate equations.

3.10 Measurement Incorporation

Measurement incorporation provides the necessary function of populating the measurement update equations described in Sections 3.11.1 - 3.11.17 with the state estimates, measurements, modeled values and other necessary parameters, and then performing the state estimate update. During the measurement update, other tasks with regard to delayed-state sensors, residual monitoring, and quaternion normalization are performed. To incorporate delayed state measurements, the delayed-state equations are modified to account for updates that occur during delayed-state measurement intervals. When no delayed-state sensors are present, the measurement update is performed using the UD update equations.

Residual monitoring is performed at every measurement update cycle. Measurement residuals that extend beyond three times the residual standard deviation are flagged and the associated measurements are deleted. Then the measurement update is performed, and the updated state estimate is saved.

3.10.1 Delayed-State Modifications. In Section 2.9.3, the delayed-state Kalman gain and state covariance update equations are formed assuming that no other updates will take place during the interval that the delayed-state measurement is valid over. That is, it is assumed that the best estimate at the time of the previous measurement is related to the propagated states as:

$$\mathbf{x}_{k-1} = \mathbf{\Phi}_{k-1}^{-1}\mathbf{x}_k^- - \mathbf{\Phi}_{k-1}^{-1}\mathbf{w}_{k-1} \tag{3.20}$$

This is not true in this project. It is highly likely that an update occurred in the interval. To account for updates that occur in the measurement interval, we return to the derivation of the delayed-state Kalman gain and updated covariance in(2.76) and (2.78) respectively. The modified delayed-state Kalman gain and covariance update is formed in [5] using the Kalman gain and state covariance update equations where it is not assumed that the measurement and process noise are independent:

$$\mathbf{K}_k = (\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{C}_k)[\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T + \mathbf{R}_k + \mathbf{H}_k\mathbf{C}_k + \mathbf{C}_k^T\mathbf{H}_k^T]^{-1} \tag{3.21}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k^-[\mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T + \mathbf{R}_k + \mathbf{H}_k\mathbf{C}_k + \mathbf{C}_k^T\mathbf{H}_k^T]\mathbf{K}_k \tag{3.22}$$

where $\mathbf{C}_k$ is a crosscorrelation parameter between measurement and process noise. To adjust these equations to the delayed-state case, the following replacements must be made in (3.21) and (3.22):

$$\mathbf{H}_k \rightarrow \mathbf{H}_k + \mathbf{J}_k\mathbf{\Phi}_{k-N}^{-1} \tag{3.23}$$

$$\mathbf{R}_k \rightarrow \mathbf{H}_k + \mathbf{J}_k\mathbf{\Phi}_{k-1}^{-1}\mathbf{Q}_{k-1}\mathbf{\Phi}_{k-N}^{-1T}\mathbf{J}_k^T \tag{3.24}$$

$$\mathbf{C}_k \rightarrow \mathbf{Q}_{k-N}\mathbf{\Phi}_{k-N}^{-1T}\mathbf{J}_k^T \tag{3.25}$$

where $\mathbf{\Phi}_{k-N}$ is the cumulative state transition matrix and $\mathbf{Q}_{k-N}$ is the accumulated process noise calculated every update cycle, and $\mathbf{H}_k$ and $\mathbf{J}_k$ are the linearized measurement equations:

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x_k}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \hat{\mathbf{x}}_{k-1}}, \ \mathbf{J}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x_{k-1}}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \hat{\mathbf{x}}_{k-1}} \tag{3.26}$$

The cumulative state transition matrix, $\mathbf{\Phi}_{k-N}$, is given by:

$$\mathbf{\Phi}_{k-N} = \prod_{M=k-N}^{k-1} \mathbf{\Phi}_M \tag{3.27}$$

and $\mathbf{Q}_{k-N}$ is given by:

$$\mathbf{Q}_{k-N} = \sum_{M=k-N}^{k-1} \mathbf{Q}_M \tag{3.28}$$

Now considering the estimated states, in order for the delayed-state equations to be valid, the updates from other sensors since the time the specific sensor gave its last measurement must be removed. In this thesis, the state change is accumulated at every measurement update and the extracted from the current best estimate when the delayed-state sensor provides its next measurement. The modified state is then used to form the measurement update equations as:

$$\mathbf{z}_k = \mathbf{h}(\hat{\mathbf{x}}_k^- - \Delta\mathbf{x}_k, \hat{\mathbf{x}}_{k-1}) + \mathbf{v}_k \tag{3.29}$$

where $\Delta\mathbf{x}$ is the accumulated state changes due to updates from the other sensors and is calculated at each time step as:

$$\Delta\mathbf{x}_k = \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^- + \Delta\mathbf{x}_{k-1} \tag{3.30}$$

Table 3.1: Sensor section reference guide.

Sensor Name	Section	Senosr Name	Section
Air Data Computer	Section 3.11.1	3-axis Magnetometer	Section 3.11.10
Barometric Altimeter	Section 3.11.2	Odometer	Section 3.11.11
Cameras	Section 3.11.3	Pseudorange/Delta Range Sensor	Section 3.11.7
Compass	Section 3.11.4	Ranging Sensor	Section 3.11.12
GPS	Section 3.11.7	RFID Device	Section 3.11.13
Inclinometer	Section 3.11.15	Signal of Opportunity Sensor	Section 3.11.14
INS	Section 3.11.6	Stop Sign Sensor	Section 3.11.15
2D Laser Scanner	Section 3.11.8	Step Sensor	Section 3.11.15
3D Laser Scanner	Section 3.11.9	Terrain Referenced Altimeter	Section 3.11.17

3.11 Sensor Measurement Models

This section contains the sensors measurement models and and update equations necessary for every sensors used in this project. Some sensors are grouped into a single Subsection if they are similar in nature (such as the monocular camera and the stereo camera pair). For many sensors, there are a variety of ways to formulate the measurement. In this thesis, one approach was chosen for each measurement type in order to be aligned with the ASPN program. The complete list of sensors discussed in the following sections are listed in Table 3.1 along with the Section number that they are discussed in.

3.11.1 Air Data Computer. The air data computer (ADC) provides measurements of true air speed. *True air speed* will from now on be referred to as *actual air speed* to avoid confusion with true versus estimated quantities. The measurement model for the actual air speed is given as:

$$V_{ADC_{meas}} = V_{ADC_{true}} + B_{wind} + w_{ADC} \tag{3.31}$$

where V_{ADC} is the actual airspeed, B_{wind} is a TCB that accounts for wind speed, and w_{ADC} is white Gaussian noise measurement error. To remove the effects of wind on the actual airspeed, it is necessary to transform the airspeed into air velocity. For

simplicity, the velocity is considered in only the horizontal dimensions which induces minimal accuracy loss for steady flights.

The actual air velocity can then be expressed in terms of the aircraft and wind velocities as:

$$\mathbf{v}_{ADC} = \mathbf{v}_{AC} - \mathbf{b}_{wind} \tag{3.32}$$

where $\mathbf{v}_{AC}$ is the aircraft velocity in the north and east channel given by:

$$\mathbf{v}_{AC} = \begin{bmatrix} v_N \\ v_E \end{bmatrix} \tag{3.33}$$

and $\mathbf{b}_{wind}$ is the two dimensional velocity of the wind modeled as two TCBs:

$$B_{wind} = \|\mathbf{b}_{wind}\| \tag{3.34}$$

3.11.1.1 Measurement Update. The ADC measurement update is a function of the aircraft velocity. The states are related to the actual air speed measurement by taking the magnitude of (3.33). The measurement update equations are then given as:

$$\begin{aligned}
\mathbf{z}_{ADC} &= h_{ADC} + w_{ADC} && (3.35)\\
z_{ADC} &= V_{ADC_{meas}} && (3.36)\\
h_{ADC} &= \|\mathbf{v}_{AC} - \mathbf{b}_{wind}\| && (3.37)\\
\mathbf{H}_{ADC} &= \left.\frac{\partial h_{ADC}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} && (3.38)\\
\mathbf{R}_{ADC} &= E[(w_{ADC})(w_{ADC})^T] && (3.39)
\end{aligned}$$

3.11.2 Barometric Altimeter Sensors. The barometric altimeters used in this research provide measurements of pressure altitude with respect to mean sea

level (MSL) or a local reference datum. The measurement is modeled as:

$$h_{meas} = h_{true} + b_{baro} + w_{baro} \tag{3.40}$$

Where h represents the altitude with respect to the reference datum, b_{baro} is a TCB process representing the slow moving height error induced from atmospheric change, height, distance from reference and other unmodeled errors, and v_{baro} represents the white Gaussian noise measurement error.

Measurements from the barometer are quantized above the measurement noise threshold with the quantization level l_{quant}. This represents a uniform distribution from $-q_{baro}/2$ to $q_{baro}/2$. This can then be approximated as Gaussian with a standard deviation:

$$\sigma_{baro}^2 = \frac{1}{12}(q_{baro})^2 \tag{3.41}$$

Finally the influence of the barometer lever arm on the measurement can be extracted by resolving the lever arm into the navigation frame and subtracting the resulting height added from the measurement as:

$$Alt = Alt_{meas} - l_D \tag{3.42}$$

where l_D is the altitude element of the baro lever arm resolved in the geodetic frame using $\mathbf{C}_n^g \mathbf{C}_b^n \mathbf{l}_{baro}^b$.

The final measurement update is then given by:

$$\begin{aligned}
z_{baro} &= h_{baro} + w_{baro} && (3.43)\\
z_{baro} &= Alt_{meas} && (3.44)\\
h_{baro} &= Alt + l_D + b_{baro} && (3.45)\\
\mathbf{H}_{baro} &= \Big[\underbrace{0 \;\; 0 \;\; 1}_{\mathbf{p}^n} \Big| \; \mathbf{0} \; \Big| \underbrace{1}_{b_{baro}} \Big| \; \mathbf{0} \; \Big]_{1\times N} && (3.46)\\
R &= E[(w_{baro})(w_{baro})^T] && (3.47)
\end{aligned}$$

3.11.3 Camera Sensors. A basic digital optical imaging camera consists of an aperture, lens, detector array and a sampling array. The imaging operation involves focusing rays of light by the lens to project the scene image onto the detector array. The detector array then converts the light energy to a voltage which is then quantized to a digital value by the sampling array [31] resulting in a digital image.

This section discusses the single and stereo camera measurements, measurement update, and feature initialization. Stereo cameras in this thesis are treated as single camera units with the exception of a more accurate feature initialization, shared feature states, and double measurements for features.

3.11.3.1 Measurement Equation. The images from each camera are preprocessed using either Scale Invariant Feature Tracking (SIFT) [17] or Speeded Up Robust Features (SURF) [3] to locate image features. Using a pin hole camera model, the feature's pixel position in the camera pixel frame is calculated in each image that it is detected in and identified with a unique feature ID that remains that remains constant in all subsequent frames. The pixel measurement is then related to a calculated feature projection using a camera lens model and optical specifications. The projection is described by the pointing vector $\mathbf{s}^c$ in the camera frame shown in Figure 3.2. The measurement presented by the camera is this vector normalized by the feature depth in the camera frame expressed as:

$$\underline{\mathbf{s}}^c = \frac{1}{s_z^c}\mathbf{s}^c,\ \mathbf{s}^c = \begin{bmatrix} s_x^c \\ s_y^c \\ s_z^c \end{bmatrix} \tag{3.48}$$

The camera measurement is then modeled as:

$$\underline{\mathbf{s}}^c_{meas} = \begin{bmatrix} \underline{\mathrm{s}}^c_{x_{meas}} \\ \underline{\mathrm{s}}^c_{y_{meas}} \end{bmatrix} = \begin{bmatrix} \underline{\mathrm{s}}^c_{x_{true}} \\ \underline{\mathrm{s}}^c_{y_{true}} \end{bmatrix} + \mathbf{w}_{cam} \tag{3.49}$$

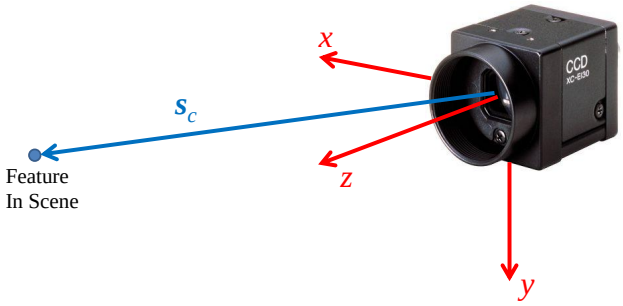

Figure 3.2: ASPN camera frame showing the projection vector, $\mathbf{s}^c$,which extends from the middle of the camera frame to the feature

Where $\underline{s}_x^c$ and $\underline{s}_y^c$ are the first two components of the homogenous pointing vector to the feature location in the camera frame, and $\mathbf{w}_{cam}$ is the white Gaussian noise measurement uncertainty. The covariance of the measurement noise is estimated during feature extraction. To relate the camera measurement to the estimated states, the feature location must be determined. The methods used in this research to determine feature location are described next.

3.11.3.2 Best Guess Feature Location Estimation. To determine the location of each new feature to be tracked, a best guess estimate is made of magnitude of the feature pointing vector in the camera frame which must then be resolved in the geodetic navigation frame. This is expressed in terms of the modified measurement as:

$$\tilde{\mathbf{s}}^c = D\bar{\mathbf{s}}_{meas}^c \tag{3.50}$$

where D is the best guess magnitude of the pointing unit vector and $\bar{\mathbf{s}}_{meas}^c$ is the normalized measurement homogenous vector given by:

$$\bar{\mathbf{s}}_{meas}^c = \frac{1}{\sqrt{1 + (\underline{s}_{meas_x}^c)^2 + (\underline{s}_{meas_x}^c)^2}} \begin{bmatrix} \underline{s}_{x_{meas}}^c \\ \underline{s}_{y_{meas}}^c \\ 1 \end{bmatrix} \tag{3.51}$$

The magnitude of the projection vector can be based off environment, terrain, camera, etc. For this thesis, the guess magnitude was dependent on the platform. The projection vector must then transformed into the geodetic navigation frame. The position of the feature in the NED frame is given as:

$$\mathbf{s}^n = \mathbf{C}_b^n[\mathbf{l}^b + \mathbf{C}_c^b\tilde{\mathbf{s}}^c] \tag{3.52}$$

where $\mathbf{l}^b$ is the camera lever arm defined in the body frame. The feature geodetic location is then given by the inverse relationship given in (2.14) where the initial location is the current platform best position estimate. These are calculated sequentially as:

$$\begin{aligned} Alt_{feat} &= Alt - s_z^n & (3.53)\\ L_{feat} &= \frac{s_x^n}{(R_p + Alt_{feat})} + L & (3.54)\\ \lambda_{feat} &= \frac{s_y^n}{(R_m + Alt_{feat})\cos L_{feat}} + \lambda & (3.55) \end{aligned}$$

where Alt is the platform altitude, L is the platform latitude, λ is the platform longitude, and s_x^n, s_y^n, and s_z^n are the three components of the estimated projection vector $\tilde{\mathbf{s}}^c$. For feature position estimation the filter must keep track of active features for the entire time the camera takes measurements. The three components of the feature's position in the geodetic frame are estimated at each filter update cycle as constant biases with a small amount of noise to allow the filter to continue making feature position corrections.

The covariance of the projection vector is given by calculating the influence matrix using:

$$\mathbf{W} = \frac{\partial \mathbf{s}^n}{\partial \mathbf{y}} \tag{3.56}$$

where $\mathbf{y}$ is given by:

$$\mathbf{y} = \begin{bmatrix} \underline{\mathbf{s}}_{meas}^c \\ D \end{bmatrix} \tag{3.57}$$

The covariance is then computed using:

$$\mathbf{P}_{\tilde{\mathbf{s}}^c\tilde{\mathbf{s}}^c} = \mathbf{W}\begin{bmatrix} E[(\mathbf{w}_{cam})(\mathbf{w}_{cam})^T] & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & P_{DD} \end{bmatrix}\mathbf{W}^T \tag{3.58}$$

where P_{DD} is the covariance of D and is set according to the platform and its environment.

3.11.3.3 Stereo Camera Feature Location Estimation. Stereo vision enables the feature distance from the camera setup to be estimated via stereopsis. This allows a much more accurate initialization of the feature position with respect to the platform body once features have been statistically matched between the two camera frames. To determine the feature distance from the camera, the relationship between the two camera measurements must be determined. From [31], the projection from a single camera to the feature is given in terms of feature's position in the NED frame as:

$$\mathbf{s}^c = \mathbf{C}_b^c(\mathbf{C}_n^b\mathbf{s}^n - \mathbf{l}_1^b) \tag{3.59}$$

Applying (3.59) to two cameras, the two projection vector equations for a single feature are given as:

$$\mathbf{s}_1^{c1} = \mathbf{C}_b^{c1}(\mathbf{C}_n^b\mathbf{s}^n - \mathbf{l}_1^b) \tag{3.60}$$

$$\mathbf{s}_2^{c2} = \mathbf{C}_b^{c2}(\mathbf{C}_n^b\mathbf{s}^n - \mathbf{l}_2^b) \tag{3.61}$$

where $\mathbf{C}_b^{c1}$ is the DCM that rotates quantities from the body frame to camera one's frame, $\mathbf{C}_b^{c2}$ is the DCM that rotates quantities from the body frame to camera two's frame, $\mathbf{l}_1^b$ is the lever arm of camera one expressed in the body frame, and $\mathbf{l}_2^b$ is the lever arm of camera two expressed in the body frame. Rearranging (3.61) to solve for the feature NED position, $\mathbf{s}^n$, and substituting into (3.60), the following relationship is obtained:

$$\mathbf{s}_1^{c1} = \mathbf{C}_{c2}^{c1}\mathbf{s}_2^{c2} + \mathbf{d},\ \mathbf{d} = \mathbf{C}_b^{c1}(\mathbf{l}_2^b - \mathbf{l}_1^b) \tag{3.62}$$

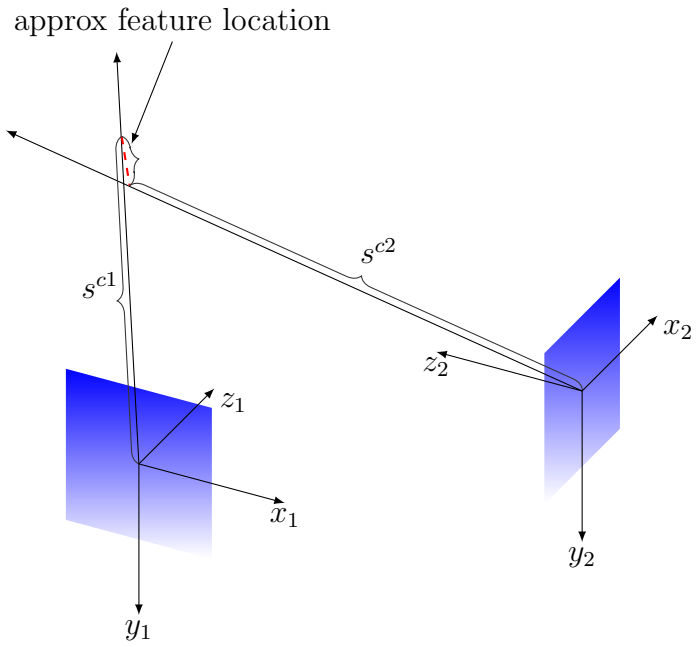

Figure 3.3: Closest point of contact between infinite extensions of camera pointing vectors. The closest the pointing vectors from camera one and camera two ever come into contact with one another is given by the end points of the red dashed line

where $\mathbf{d}$ is the vector joining the origins of the two camera frames and $\mathbf{C}_{c2}^{c1}$ is the DCM that rotates quantities from the frame of camera two to the frame of camera one. Choosing the frame of camera one to be the frame of reference, it is now desired to find the location where each projection vector meets.

Considering the physical interpretation of the pointing vectors from each camera, the magnitude for each vector is given by the scaling parameters that causes the two to meet. However, given the errors in the measurement model, it is highly unlikely that the lines will ever intersect in three dimensional space. Thus, the approximate solution is given by considering the point of closest contact between the three dimensional vectors seen in Figure 3.3. To find this point, we express the measured homogeneous pointing vectors in terms of their unit vectors $\bar{\mathbf{s}}_{1_{meas}}^{c1}$ and $\bar{\mathbf{s}}_{2_{meas}}^{c1}$ from (3.51), and the scaling parameters D_1 and D_2. The projection vector from camera one in camera one's reference frame to the feature is given as:

$$\tilde{\mathbf{s}}_1^{c1} = \bar{\mathbf{s}}_{1_{meas}}^{c1} D_1 \tag{3.63}$$

The projection vector from camera two expressed in the frame of camera one is taken from (3.62), and given as:

$$\tilde{\mathbf{s}}_2^{c1} = \mathbf{d} + \mathbf{C}_{c2}^{c1}\bar{\mathbf{s}}_{2_{meas}}^{c2} D_2 \tag{3.64}$$

The appropriate values for D_1 and D_2 are found by minimizing the gradient of the distance between every point of the three dimensional line formed by varying the scaling parameters of (3.63) and (3.64) [9]. The result is given in [28] as:

$$D_1 = \frac{be - cd}{ac - b^2}, \text{ and } D_2 = \frac{ae - bd}{ac - b^2} \tag{3.65}$$

where

$$a = \bar{\mathbf{s}}_{1_{meas}}^{c1} \cdot \bar{\mathbf{s}}_{1_{meas}}^{c1} \tag{3.66}$$

$$b = \bar{\mathbf{s}}_{1_{meas}}^{c1} \cdot (\mathbf{C}_{c2}^{c1}\bar{\mathbf{s}}_{2_{meas}}^{c2}) \tag{3.67}$$

$$c = (\mathbf{C}_{c2}^{c1}\bar{\mathbf{s}}_{2_{meas}}^{c2}) \cdot (\mathbf{C}_{c2}^{c1}\bar{\mathbf{s}}_{2_{meas}}^{c2}) \tag{3.68}$$

$$d = \bar{\mathbf{s}}_{1_{meas}}^{c1} \cdot \mathbf{d} \tag{3.69}$$

$$e = (\mathbf{C}_{c2}^{c1}\bar{\mathbf{s}}_{2_{meas}}^{c2}) \cdot \mathbf{d} \tag{3.70}$$

The denominator of (3.65), will only equal zero when the lines are parallel giving a feature location at infinity, and it will only be less than zero when the lines do not converge in front of the camera. This situation will only occur as a noise induced error and can be easily monitored. If either of these situations are detected then the feature is discarded. The approximate feature projection vector is then calculated using the mean of the projection feature from each camera:

$$\tilde{\mathbf{s}}^c = \frac{1}{2}(\tilde{\mathbf{s}}_2^{c1} - \tilde{\mathbf{s}}_1^{c1}) + \tilde{\mathbf{s}}_1^{c1} \tag{3.71}$$

The resulting NED position is then found using (3.52) which is repeated here for convenience:

$$\mathbf{s}^n = \mathbf{C}_b^n[\mathbf{l}^b + \mathbf{C}_c^b\tilde{\mathbf{s}}^c] \tag{3.72}$$

3.11.3.4 Feature Location Covariance. The initial feature covariance is based on the uncertainty in the current platform position coupled with the uncertainty in the initial feature position guess with respect to the camera frame, $\mathbf{s}^c$. The covariance can be estimated by forming the Jacobian with the partials of (3.52) with respect to the position states and the components of the initial feature position in the camera frame as:

$$\mathbf{W} = \frac{\partial \mathbf{t}^n}{\partial \mathbf{y}} \tag{3.73}$$

where $\mathbf{y}$ and its associated covariance, $\mathbf{P_{yy}}$, are given as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{s}}^c \end{bmatrix} \mathbf{P_{yy}} = \begin{bmatrix} \mathbf{P_{xx}} & \mathbf{0} \\ \mathbf{0} & \mathbf{P_{\tilde{s}^c\tilde{s}^c}} \end{bmatrix} \tag{3.74}$$

where $\mathbf{x}$ is the state vector of all the estimated states, $\mathbf{P_{xx}}$ is the navigation states covariance matrix, $\tilde{\mathbf{s}}^c$ is the estimated feature projection vector, $\mathbf{P_{\tilde{s}^c\tilde{s}^c}}$ is the estimated feature projection vector covariance matrix, and $\mathbf{t}^n$ is the feature position in the geodetic frame given by (3.53) - (3.55).

The associated error mean and covariance value of the feature position is then given by:

$$\delta \mathbf{t}^n = \mathbf{W}\mathbf{y} \tag{3.75}$$

$$E[(\delta \mathbf{t}^n)(\delta \mathbf{t}^n)^T] = \mathbf{W}\mathbf{P_{yy}}\mathbf{W}^T \tag{3.76}$$

The feature position is correlated with the platform states, and the cross-covariance terms for every previously estimated state must be calculated. These terms are given by:

$$\mathbf{P}_{\delta \mathbf{t}^n \delta \mathbf{x}} = E[(\delta \mathbf{t}^n)(\mathbf{x})^T] \tag{3.77}$$

Substituting (3.75) into (3.77), the cross-covariance terms are obtained by:

$$\begin{aligned} \mathbf{P}_{\delta\mathbf{t}^n\mathbf{x}} &= E[(\mathbf{W}\mathbf{y})(\mathbf{x})^T] \\ &= \mathbf{W}E[(\mathbf{y})(\mathbf{x})^T] \end{aligned} \tag{3.78}$$

because we know that the last three elements of $\mathbf{y}$ are independent of $\mathbf{x}$, (3.78) can be rewritten as:

$$\mathbf{P}_{\delta\mathbf{t}^n\delta\mathbf{x}} = \mathbf{W}'E[(\mathbf{x})(\mathbf{x})^T] \tag{3.79}$$

where $\mathbf{W}'$ is equal to $\mathbf{W}$ with the last 3×3 matrix element removed. The vertical cross-covariance terms are then given by:

$$\mathbf{P}_{\delta\mathbf{t}^n\mathbf{x}} = \mathbf{W}'\mathbf{P}_{\mathbf{xx}} \tag{3.80}$$

and equivalently the horizontal cross-covariance terms are given by:

$$\mathbf{P}_{\mathbf{x}\delta\mathbf{t}^n} = \mathbf{P}^T_{\delta\mathbf{t}^n\mathbf{x}} \tag{3.81}$$

The cross-covariance terms and feature location covariance must then be carefully placed into the proper locations in the state covariance matrix.

The covariance of the projection vector is given using (3.58) where P_{DD} is the covariance of D and can be calculated by either tracking the measurement covariance through (3.63) - (3.71) or setting this to an artificially high value so that the filter converges on an estimate provided by measurement updates.

3.11.3.5 Camera Measurement Update. For the monocular camera, the measurement is then expressed in terms of the states using the projection model as:

$$\mathbf{s}^c = \mathbf{C}^c_b(\mathbf{C}^b_n\mathbf{s}^n - \mathbf{l}^b) \tag{3.82}$$

where $\mathbf{s}^n$ is the feature position in the NED frame using the relationship to the geodetic frame in (2.14) as:

$$\mathbf{s}^n = \begin{bmatrix} (R_p + Alt_{feat})(L_{feat} - L) \\ (R_m + Alt_{feat}) \cos L(\lambda_{feat} - \lambda) \\ Alt - Alt_{feat} \end{bmatrix} \tag{3.83}$$

Finally, (3.82) is normalized by the z component of the projection as in (3.48). The final measurement update equations are found as:

$$\mathbf{z}_{cam} = \mathbf{h}_{cam} + \mathbf{w}_{cam} \tag{3.84}$$

$$\mathbf{z}_{cam} = \begin{bmatrix} \underline{s}^c_{x_{meas}} \\ \underline{s}^c_{y_{meas}} \end{bmatrix} \tag{3.85}$$

$$\mathbf{h}_{cam} = \frac{1}{s^c_z}\mathbf{s}^c \tag{3.86}$$

$$\mathbf{H}_{cam} = \left.\frac{\partial \mathbf{h}_{cam}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.87}$$

$$\mathbf{R}_{cam} = E[(\mathbf{w}_{cam})(\mathbf{w}_{cam})^T] \tag{3.88}$$

The measurement update for the stereo camera pair follows the same method as with single cameras but uses two measurements for a single feature instead of one. The respective lever arms and camera orientations must be taken into consideration in (3.82) when forming the respective measurement equations.

3.11.4 Compass Sensors. The earth's magnetic field generally runs parallel to the earth's surface and points toward magnetic north. A compass exploits the ubiquitous magnetic field to provide measurements of heading. Compass measurements are affected by numerous entities including shock and vibration, stray magnetic fields, and unlevel supporting surfaces [36]. These errors are modeled in this thesis as a single TCB bias. The measurement equation for the compass is given as:

$$\psi_{meas} = \psi_{true} + b_\psi + w_\psi \tag{3.89}$$

Where ψ is the heading angle, b_ψ is a TCB bias that accounts for the common compass errors, and w_ψ is the white Gaussian noise measurement error.

3.11.4.1 Measurement Update. The heading angle is related to the quaternion states via the four quadrant inverse tangent:

$$\psi = \tan^{-1}(c_{21}/c_{11}) \tag{3.90}$$

Where c_{11} and c_{21} are the respective elements taken from the body to NED frame DCM given in (3.15) with the corresponding quaternion elements seen in (2.15). The compass lever arm has a negligible effect on the heading measurement. The measurement update equation is then given as:

$$\begin{aligned}
z_{comp} &= h_{comp} + w_\psi && (3.91)\\
z_{comp} &= \psi_{meas} && (3.92)\\
h_{comp} &= \tan^{-1}\left(\frac{2(q_2q_3 - q_1q_4)}{q_1^2 + q_2^2 - q_3^2 - q_4^2}\right) + b_\psi && (3.93)\\
\mathbf{H}_{comp} &= \left.\frac{\partial h_{comp}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} && (3.94)\\
\mathbf{R}_{comp} &= E[(w_\psi)(w_\psi)^T] && (3.95)
\end{aligned}$$

3.11.5 Inclinometer. An inclinometer provides measurements of tilt of the surface that it rests on. For a platform, this equates to providing information about pitch and roll. Inclinometers provide very useful information about attitude and can be used to aid in estimating compass errors [36]. Inclinometers performance generally suffers from errors caused by its own motion [18]. These errors are accounted for in this thesis using TCB biases in the pitch and roll measurements. The inclinometer measurement model is given as:

$$\begin{bmatrix} \theta_{meas} \\ \phi_{meas} \end{bmatrix} = \begin{bmatrix} \theta_{true} \\ \phi_{true} \end{bmatrix} + \mathbf{b}_{incl} + \mathbf{w}_{incl} \tag{3.96}$$

Where θ and ϕ are the pitch and roll angles, $\mathbf{b}_{incl}$ is a TCB bias for each quantity and $\mathbf{w}_{incl}$ is the measurement white Gaussian noise error.

3.11.5.1 Measurement Update. These measurements are incorporated into the filter by relating the pitch and roll measurements to the quaternion states through (3.15) and (2.15). Like the compass, the lever arm is also negligible to this measurement. The resulting measurement update equations then take the form:

$$\mathbf{z}_{incl} = \mathbf{h}_{incl} + \mathbf{w}_{incl} \tag{3.97}$$

$$\mathbf{z}_{incl} = \begin{bmatrix} \theta_{meas} \\ \phi_{meas} \end{bmatrix} \tag{3.98}$$

$$\mathbf{h}_{incl} = \begin{bmatrix} \sin^{-1}(2(q_2q_4 - q_1q_3) \\ \tan^{-1}\left(\frac{2(q_3q_4 + q_1q_2)}{q_1^2 - q_2^2 - q_3^2 + q_4^2}\right) \end{bmatrix} + \mathbf{b}_{incl} \tag{3.99}$$

$$\mathbf{H}_{incl} = \left.\frac{\partial \mathbf{h}_{incl}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.100}$$

$$\mathbf{R}_{incl} = E[(\mathbf{w}_{incl})(\mathbf{w}_{incl})^T] \tag{3.101}$$

3.11.6 Inertial Measurement Units. An inertial navigation unit (INS) or equivalently an IMU, measures the specific force acting in the sensor frame. Generally this will be coincident with the body frame or be a fixed orientation with respect to the body frame. The measured specific force includes not only the body accelerations, but also the gravitational force acting upon it. These the two forces can be differentiated only by knowing the body current attitude and modeling the gravitational force.

There are two IMU measurement types – sampled specific force and angular rate and integrated specific force and angular rate. The noise model and treatment of the sensor in the Kalman filter is will differ depending on the measurement type. Error terms are valid along all three axes of the IMU sensor frame and noise characteristics are described using scalar math on only one axis for clarity.

3.11.6.1 IMU Sampled Specific Force Measurement. The sampled specific force measurement presented by an IMU in this thesis is modeled by:

$$\mathbf{f}_{meas} = \mathbf{f}_{true} + \mathbf{b}_a + \mathbf{w}_a \tag{3.102}$$

Where $\mathbf{f}_{true}$ is the true specific force, $\mathbf{b}_a$ is a TCB bias error, and $\mathbf{w}_a$ represents the process white Gaussian noise. The process noise autocorrelation along the x-axis and derived noise power Q_{v_a} is given by

$$E[w_{a_x}(t_j)w_{a_x}(t_k)] = Q_{w_a}\delta_{jk} \tag{3.103}$$

$$Q_{w_a} = \sigma^2_{w_{a_x}} \tag{3.104}$$

The TCB bias, $\mathbf{b}_a$, noise power is given by (2.35) using given IMU time constant and bias variance. The physical relationship between the specific force process noise and the IMU manufacturer parameters are related through the velocity random walk term VRW. The associated measurement variance can then be calculated using the relationship

$$\sigma^2_{w_{a_x}} = (VRW)^2/\Delta t \tag{3.105}$$

where Δt is the time interval between measurements.

3.11.6.2 Integrated Specific Force IMU Measurement. The integrated specific force measurement equation differs slightly for the specific force measurement equation and is modeled by:

$$\Delta\mathbf{v}_{meas} = \Delta\mathbf{v}_{true} + \mathbf{b}_{\Delta v} + \mathbf{w}_{\Delta v} \tag{3.106}$$

Where $\mathbf{v}_{true}$ is the true change in velocity over the past time period, $\mathbf{b}_{\Delta v}$ is a TCB bias error and $\mathbf{w}_{\Delta v}$ represents the measurement white Gaussian noise error. The process

noise autocorrelation along the x-axis and derived noise power $Q_{w_{\Delta v}}$ is given by:

$$E[w_{\Delta v_x}(t_j)w_{\Delta v_x}(t_k)] = Q_{w_{\Delta v}}\delta_{jk} \tag{3.107}$$

$$Q_{w_{\Delta v}} = \sigma^2_{w_{\Delta v_x}} \tag{3.108}$$

The TCB bias noise power is given by (2.35) and calculated as:

$$Q_{b_{w_{\Delta v}}} = 2\frac{\sigma^2_{b_{w_{\Delta v}}}\Delta t}{T_{b_{w_{\Delta v}}}} \tag{3.109}$$

The variance of the measurement is calculated using the VRW parameter:

$$\sigma^2_{w_{\Delta v_x}} = (VRW)^2\Delta t \tag{3.110}$$

3.11.6.3 IMU Sampled Angular Rate Gyro Measurement. The sampled angular rate measurement presented by an IMU is modeled as:

$$\boldsymbol{\omega}^b_{ib_{meas}} = \boldsymbol{\omega}^b_{ib_{true}} + \mathbf{b}_\omega + \mathbf{w}_\omega \tag{3.111}$$

Where $\boldsymbol{\omega}^b_{ib_{true}}$ is the true angular rate, $\mathbf{b}_\omega$ is a TCB bias error and $\mathbf{w}_\omega$ represents the process white Gaussian noise. The process noise autocorrelation along the x-axis and derived noise power Q_{v_ω} is given by:

$$E[w_{\omega_x}(t_j)w_{\omega_x}(t_k)] = \sigma^2_{w_{\omega_x}}\delta_{jk} \tag{3.112}$$

$$Q_{w_\omega} = \sigma^2_{w_{\omega_x}} \tag{3.113}$$

The TCB bias, $\mathbf{b}_\omega$, noise power is given by (2.35) using given IMU time constant and bias variance. The angular rate measurement variance is given by the physical accuracy parameter, angular random walk ARW through the relationship:

$$\sigma^2_{w_{\omega_x}} = (ARW)^2/\Delta t \tag{3.114}$$

where Δt is the time interval between measurements.

3.11.6.4 IMU Integrated Angular Rate Gyro Measurement. The integrated angular rate measurement equation is given by:

$$\Delta\boldsymbol{\theta}_{meas} = \Delta\boldsymbol{\theta}_{true} + \mathbf{b}_{\Delta\theta} + \mathbf{w}_{\Delta\theta} \tag{3.115}$$

where $\boldsymbol{\theta}_{true}$ is the true change in velocity over time period since the last measurement, $\mathbf{b}_{\Delta\theta}$ is a TCB bias error and $\mathbf{w}_{\Delta\theta}$ represents the measurement white Gaussian noise error. The process noise autocorrelation along the x-axis and derived noise power $Q_{v_{\Delta\theta}}$ is given by:

$$E[w_{\Delta\theta_x}(t_j)w_{\Delta\theta_x}(t_k)] = Q_{w_{\Delta\theta}}\delta_{jk} \tag{3.116}$$

$$Q_{w_{\Delta\theta}} = \sigma^2_{w_{\Delta\theta_x}} \tag{3.117}$$

The TCB bias, $\mathbf{b}_{\Delta\theta}$, noise power is given by (2.35) using given IMU time constant and bias variance multiplied by the time since the last measurement which is:

$$Q_{b_{\Delta\theta}} = 2\frac{\sigma^2_{b_{\Delta\theta}}\Delta t}{T_{b_{\Delta\theta}}} \tag{3.118}$$

The variance of the measurement noise is calculated using the ARW parameter:

$$\sigma^2_{w_{\Delta\theta_x}} = (ARW)^2\Delta t \tag{3.119}$$

3.11.6.5 IMU Measurement Type Treatment. The two IMU measurement types described in the previous sections require separate treatment when incorporated into the filter. A rigorous incorporation would include treating accumulated acceleration, and angular rate measurements as delayed-state quantities. This treatment was undesirable for multiple reasons including complexity, time of implementation, and numerical accuracy issues. The first two reason are somewhat obvious,

but the third reason deserves deeper discussion. The Kalman filter delayed-state equations are not available in literature in UD factorization form. Because IMU measurements are available at a high frequency, they provide the majority of the filter update and propagation times. Treating an accumulated acceleration/angular rate IMU as a delayed-state sensor would require significantly more updates and propagations to not be performed using the numerically stable UD equations.

In order to avoid a delayed-state representation, the accumulated acceleration/angular rate IMU was then handled by converting the measurements into instantaneous acceleration and angular rate by dividing the measurements by the integration interval. The measurement time was considered to the point midway in the integration. In practice, this would require a minimal filter delay since measurements would be available at half a time epoch after they should have been incorporated. Because of the high measurement frequency, this would not be a significant real time performance degradation on a real system, but would have caused a significant complexity burden on the filter code built for this project. Thus accumulated acceleration/angular rate IMU measurements are preprocessed by moving measurements backward in time one half epoch as the measurements are buffered from a binary data file.

3.11.6.6 Acceleration Measurement Update. For this project, both INS types were treated essentially the same with the measurement equation (3.102). Expressing (3.102) with more descriptive annotation, the IMU acceleration measurement is:

$$\mathbf{f}^b_{meas} = \mathbf{f}^b_{true} + \mathbf{b}_a + \mathbf{w}_a \tag{3.120}$$

where $\mathbf{f}^b$ is the specific force in the body frame and can be related to the body acceleration and physical geographic quantities from [29] as:

$$\mathbf{f}^b = \mathbf{a}^b + \mathbf{C}^b_n([2\mathbf{\Omega}^n_{ie} + \mathbf{\Omega}^n_{en}]\mathbf{v}^n - \mathbf{g}^n) \tag{3.121}$$

where $\mathbf{\Omega}_{ie}^{n}$ is the inertial to ECEF frame skew symmetric rotation rate expressed in the NED frame calculated using (2.3) and (2.10) given in (3.122), and $\mathbf{\Omega}_{en}^{n}$ is the ECEF to NED skew symmetric rotation rate expressed in the NED frame given in (2.5).

$$\mathbf{\Omega}_{ie}^{n} = \mathbf{C}_{e}^{n}\mathbf{\Omega}_{ie}^{e}\mathbf{C}_{n}^{e} \tag{3.122}$$

The lever arm affects the measurement by introducing an additional acceleration caused by angular rotations about the platform COG. This additional acceleration is tangent to the estimated state acceleration $\mathbf{a}^{b}$, and is given by the double cross product of the angular rate and the lever arm. Substituting (3.121) into (3.120) and adding the lever arm effects, the total measured specific force is then given as:

$$\mathbf{f}^{b} = \mathbf{a}^{b} + \mathbf{C}_{n}^{b}([2\mathbf{\Omega}_{ie}^{n} + \mathbf{\Omega}_{en}^{n}]\mathbf{v}^{n} - \mathbf{g}^{n}) + (\mathbf{\Omega}_{ib}^{b})^{2}\mathbf{l}^{b} \tag{3.123}$$

where $\mathbf{l}^{b}$ is the IMU lever arm in the body frame. The following accelerometer measurement update equations are then presented to the filter:

$$\begin{aligned}
\mathbf{z}_{ins_a} &= \mathbf{h}_{ins_a} + \mathbf{w}_{a} && (3.124)\\
\mathbf{z}_{ins_a} &= \mathbf{f}_{meas}^{b} && (3.125)\\
\mathbf{h}_{ins_a} &= \mathbf{a}^{b} + \mathbf{C}_{n}^{b}([2\mathbf{\Omega}_{ie}^{n} + \mathbf{\Omega}_{en}^{n}]\mathbf{v}^{n} - \mathbf{g}^{n}) + (\mathbf{\Omega}_{ib}^{b})^{2}\mathbf{l}^{b} + \mathbf{b}_{a} && (3.126)\\
\mathbf{H}_{ins_a} &= \left.\frac{\partial \mathbf{h}_{ins_a}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} && (3.127)\\
\mathbf{R}_{ins_a} &= E[(\mathbf{w}_{a})(\mathbf{w}_{a})^{T}] && (3.128)
\end{aligned}$$

3.11.6.7 Angular Rate Measurement Update. The measured angular rate is not affected by the lever arm. The angular rate is between in the inertial and body frame. The gyro measurement is related to the estimated states by the body angular rate combined with the angular rate induced by motion over a curved surface and the earth rotation rate [29] as:

$$\boldsymbol{\omega}_{ib_{meas}}^{b} = \boldsymbol{\omega}_{nb}^{b} + \mathbf{C}_{n}^{b}[\boldsymbol{\omega}_{en}^{n} + \mathbf{C}_{e}^{n}\boldsymbol{\omega}_{ie}^{e}] + \mathbf{b}_{\omega} + \mathbf{w}_{\omega} \tag{3.129}$$

The measurement update equations presented to the filter are given as:

$$\mathbf{z}_{ins_\omega} = \mathbf{h}_{ins_\omega} + \mathbf{w}_\omega \tag{3.130}$$

$$\mathbf{z}_{ins_\omega} = \boldsymbol{\omega}^b_{ib_{meas}} \tag{3.131}$$

$$\mathbf{h}_{ins_\omega} = \boldsymbol{\omega}^b_{nb} + \mathbf{C}^b_n[\boldsymbol{\omega}^n_{en} + \mathbf{C}^n_e\boldsymbol{\omega}^e_{ie}] + \mathbf{b}_\omega \tag{3.132}$$

$$\mathbf{H}_{ins_\omega} = \left[\begin{array}{c|c|c|c|c} \mathbf{0} & \underbrace{\mathbf{I}_3}_{\boldsymbol{\omega}^b_{nb}} & \mathbf{0} & \underbrace{\mathbf{I}_3}_{\mathbf{b}_\omega} & \mathbf{0} \end{array}\right]_{3\times N} \tag{3.133}$$

$$\mathbf{R}_{ins_\omega} = E[(\mathbf{w}_\omega)(\mathbf{w}_\omega)^T] \tag{3.134}$$

3.11.7 GNSS Sensors. Global navigation satellite system (GNSS) sensors is a generic term used in this thesis to encapsulate GPS position and/or velocity measurements and GPS sensors that give pseudorange/delta-range measurements. Position measurements are expressed using the WGS-84 geodetic latitude, longitude, and altitude quantities. When available from measuring the doppler shift of the GPS carrier frequency [35], the velocity measurements are given in the NED frame. GNSS pseudorange/delta-range sensors pseudorange measurements are corrected for known and modeled errors including tropospheric delay and satellite clock offset. Delta-range measurements are corrected for known and modeled errors including the change in clock offset and changes in the tropospheric delay.

3.11.7.1 GPS Position Measurement. The position measurement of the GPS sensor is modeled by:

$$\begin{bmatrix} L_{meas} \\ \lambda_{meas} \\ h_{meas} \end{bmatrix} = \begin{bmatrix} L_{true} \\ \lambda_{true} \\ h_{true} \end{bmatrix} + \mathbf{w}_{GPS_{pos}} \tag{3.135}$$

where L is the geodetic latitude, λ is longitude, Alt is the altitude, and $\mathbf{w}_{GPS_{pos}}$ is given by:

$$\mathbf{w}_{GPS_{pos}} = \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}_{GPS_{pos}} \tag{3.136}$$

where R_E is the radius of the earth, and $\boldsymbol{\nu}_{GPS_{pos}}$ is the white Gaussian noise measurement position error in meters. The velocity measurement model is given as:

$$\mathbf{v}^n_{meas} = \mathbf{v}^n_{true} + \mathbf{w}_{GPS_{vel}} \tag{3.137}$$

Where $\mathbf{v}^n_{meas}$ is the velocity measurement in the NED reference frame and $\mathbf{w}_{GNSS_{vel}}$ is the white Gaussian noise measurement error. The measurement noise power can be estimated by the sensor as measurements are taken or be user defined. The GPS lever arm gives a spatial displacement to the GPS position measurement as:

$$\mathbf{p}^n_{GPS} = \begin{bmatrix} L \\ \lambda \\ h \end{bmatrix} + \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}^g_n \mathbf{C}^n_b \mathbf{l}^b \tag{3.138}$$

The lever arm only affects the velocity measurement during turns. The effect is considered negligible in this research and not pursued further.

The position and velocity measurements are related directly to the estimated states with the resulting measurement update equations given as:

$$\mathbf{z}_{GPS} = \mathbf{h}_{GPS} + \begin{bmatrix} \mathbf{w}_{GPS_{pos}} \\ \mathbf{w}_{GPS_{vel}} \end{bmatrix} \tag{3.139}$$

$$\mathbf{z}_{GPS} = \begin{bmatrix} L_{meas} & \lambda_{meas} & Alt_{meas} & (\mathbf{v}^n_{meas})^T \end{bmatrix}^T \tag{3.140}$$

$$\mathbf{h}_{GPS} = \begin{bmatrix} (\mathbf{p}^n_{GPS})^T & (\mathbf{v}^n)^T \end{bmatrix}^T \tag{3.141}$$

$$\mathbf{H}_{GPS_{pos}} = \begin{bmatrix} \underbrace{\mathbf{I}_3}_{L\lambda Alt} & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \end{bmatrix}_{3\times N} \tag{3.142}$$

$$\mathbf{H}_{GPS_{vel}} = \begin{bmatrix} \mathbf{0}_3 & \underbrace{\mathbf{I}_3}_{\mathbf{v}^n} & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \end{bmatrix}_{3\times N} \tag{3.143}$$

$$\mathbf{H}_{GPS} = \begin{bmatrix} \mathbf{H}_{GPS_{pos}} \\ \mathbf{H}_{GPS_{vel}} \end{bmatrix} \tag{3.144}$$

$$\mathbf{R}_{GPS} = \begin{bmatrix} E[(\mathbf{w}_{GPS_{pos}})(\mathbf{w}_{GPS_{pos}})^T] & 0 \\ 0 & E[(\mathbf{w}_{GPS_{vel}})(\mathbf{w}_{GPS_{vel}})^T] \end{bmatrix} \tag{3.145}$$

3.11.7.2 GNSS Pseudorange/Delta-range Sensors. The GNSS pseudorange/delta-range sensors measurement is modeled as:

$$\begin{bmatrix} r_{meas} \\ \Delta r_{meas} \end{bmatrix} = \begin{bmatrix} r_{true} \\ \Delta r_{true} \end{bmatrix} + \begin{bmatrix} c\delta t_r \\ c\delta t_{\Delta r} \end{bmatrix} + \begin{bmatrix} w_r \\ w_{\Delta r} \end{bmatrix} \tag{3.146}$$

where $c\delta t_r$ is the clock error bias in the range measurement and $c\delta t_{\Delta r}$ is the change in the clock error bias over the time interval that Δr is valid. The GNSS pseudorange/delta-range sensors pseudorange measurement equation is the familiar GPS range equation and is related to the estimated states by:

$$r = \|\mathbf{SV} - \mathbf{p}^e\| + c\delta t_r \tag{3.147}$$

where $\mathbf{SV}$ is the satellite ECEF location and $\mathbf{p}^e$ is given by:

$$p_x^e = (R_p + h)\cos L_{PRDR}\cos\lambda_{PRDR} \tag{3.148}$$

$$p_y^e = (R_p + h)\cos L_{PRDR}\sin\lambda_{PRDR} \tag{3.149}$$

$$p_z^e = (R_p(1-e^2) + h_{PRDR})\sin L_{PRDR} \tag{3.150}$$

and where

$$\begin{bmatrix} L_{PRDR} \\ \lambda_{PRDR} \\ h_{PRDR} \end{bmatrix} = \begin{bmatrix} L \\ \lambda \\ h \end{bmatrix} + \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E\cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_n^g \mathbf{C}_b^n \mathbf{l}^b \tag{3.151}$$

The delta-range measurement is obtained by the integrated doppler. It is a delayed-state measurement and is given by the range change over a specified time interval t_{k-1} to t_k, which does not necessarily reference the same timing of the pseudorange measurement. The measurement update is then expressed as:

$$\Delta r = r_k - r_{k-1} + c\delta t_{\Delta r} \tag{3.152}$$

where $\delta t_{\Delta r}$ is the difference in clock error between t_{k-1} and t_k, and the relationship to the state is given by (3.147). Converting the ECEF position to the NED frame using the inverse of the relationship found in (3.148), the measurement update equations presented to the filter for a single pseudorange/delta range measurement take on the

form:

$$\mathbf{z}_{PRDR} = \mathbf{h}_{PRDR} + \begin{bmatrix} w_r \\ w_{\Delta r} \end{bmatrix} \tag{3.153}$$

$$\mathbf{z}_{PRDR} = \begin{bmatrix} r_{meas} \\ \Delta r_{meas} \end{bmatrix} \tag{3.154}$$

$$\mathbf{h}_{PRDR} = \begin{bmatrix} \|\mathbf{SV} - \mathbf{p}^e\| + c\delta t_r \\ r_k - r_{k-1} + c\delta t_{\Delta r} \end{bmatrix} \tag{3.155}$$

$$\mathbf{H}_{PRDR} = \left.\frac{\partial \mathbf{h}_{PRDR}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.156}$$

$$\mathbf{R}_{PRDR} = \begin{bmatrix} E[(w_r)(w_r)^T] & 0 \\ 0 & E[(w_{\Delta r})(w_{\Delta r})^T] \end{bmatrix} \tag{3.157}$$

3.11.8 2D Laser Scanners. The laser scanners referred to in this project are range finding LiDAR or LADAR systems. The range measurements are based off of round trip time of the scanning laser signal. Given the round trip time of pulses of light with the scan angle, a detailed map of the observed environment can be generated.

Multiple methods exist for transforming the observed LiDAR data into position guidance information. The method employed in this research for both the $2D$ LiDAR systems involve processing the measurements into horizontal changes in position and heading angle change. Using these measurements of relative position allow the navigation system to bound the drift error of the onboard INS [2].

The 2D Laser scanners provide two dimensional scans of the environment around them. The 2D laser scanner measurement reference frame follows an ENU body convention as shown in Figure 3.4. For the scans, stationary environments are assumed. The preprocessing of the scans involves converting the measurements from the scanner

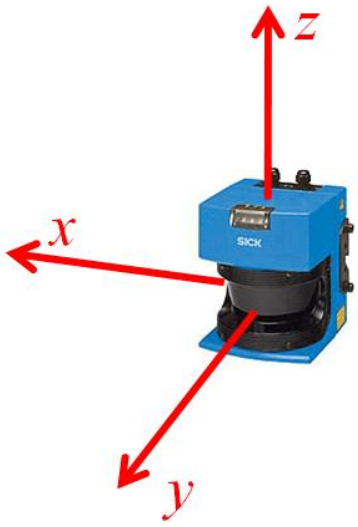

Figure 3.4: ASPN 2D Laser Frame

directly into position changes. This measurement equation is then expressed as:

$$\begin{bmatrix} \Delta x_{meas} \\ \Delta y_{meas} \\ \Delta \psi_{meas} \end{bmatrix} = \begin{bmatrix} \Delta x_{true} \\ \Delta y_{true} \\ \Delta \psi_{true} \end{bmatrix} + \mathbf{w}_{L2} \tag{3.158}$$

Where Δx and Δy are the x and y position change in the body frame, $\Delta \psi$ is the yaw angle change, and $\mathbf{w}_{L2}$ is the measurement white Gaussian noise.

3.11.8.1 Measurement Update. The measurement update equation is related to the states through the delayed-state Kalman filter equations. The position measurement is a function of the estimated NED position states from the current and past epoch where the last measurement was taken, and the quaternion states. The direct relationship can be found directly from the geodetic position using (2.14) with the previous position as the origin as:

$$\Delta \mathbf{p}^s = \mathbf{C}_b^s \mathbf{C}_n^b \left(\begin{bmatrix} (R_p + Alt_k)(L_k - L_{k-1}) \\ (R_m + Alt_k) \cos L_k (\lambda_k - \lambda_{k-1}) \\ Alt_{k-1} - Alt_k \end{bmatrix} + \Delta \mathbf{C}_b^n \mathbf{l}^b \right) \tag{3.159}$$

where $\Delta\mathbf{C}_b^n$ represents the attitude change from the last 2D laser update DCM, and $\mathbf{l}^b$ is the 2D laser scanner lever arm in the body frame. The heading measurement relates directly to the quaternion states from the heading relationship in (2.19) using the quaternion compositions of the body to NED DCM elements as:

$$\Delta\psi = \tan^{-1}(c_{21_k}/c_{11_k}) - \tan^{-1}(c_{21_{k-1}}/c_{11_{k-1}}) \tag{3.160}$$

where

$$\begin{aligned} c_{11_k} &= q_{1_k}^2 + q_{2_k}^2 - q_{3_k}^2 - q_{4_k}^2 && (3.161)\\ c_{21_k} &= q_{2_k}q_{3_k} - q_{1_k}q_{4_k} && (3.162)\\ c_{11_{k-1}} &= q_{1_{k-1}}^2 + q_{2_{k-1}}^2 - q_{3_{k-1}}^2 - q_{4_{k-1}}^2 && (3.163)\\ c_{21_{k-1}} &= q_{2_{k-1}}q_{3_{k-1}} - q_{1_{k-1}}q_{4_{k-1}} && (3.164) \end{aligned}$$

The final measurement is composed of the first two elements of (3.159) and the heading change measurement. These are then presented to the filter as:

$$\begin{aligned} \mathbf{z}_{L2} &= \mathbf{h}_{L2} + \mathbf{w}_{L2} && (3.165)\\ \mathbf{z}_{L2} &= \begin{bmatrix} \Delta x_{meas} \\ \Delta y_{meas} \\ \Delta\psi_{meas} \end{bmatrix} && (3.166)\\ \mathbf{h}_{L2} &= \begin{bmatrix} (R_p + Alt_k)(L_k - L_{k-1}) \\ (R_m + Alt_k)\cos L_k(\lambda_k - \lambda_{k-1}) \\ \tan^{-1}(c_{21_k}/c_{11_k}) - \tan^{-1}(c_{21_{k-1}}/c_{11_{k-1}}) \end{bmatrix} && (3.167)\\ \mathbf{H}_{L2} &= \left.\frac{\partial\mathbf{h}_{L2}}{\partial\mathbf{x_k}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_{k-1}} && (3.168)\\ \mathbf{J}_{L2} &= \left.\frac{\partial\mathbf{h}_{L2}}{\partial\mathbf{x_{k-1}}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_{k-1}} && (3.169)\\ \mathbf{R}_{L2} &= E[(\mathbf{w}_{L2})(\mathbf{w}_{L2})^T] && (3.170) \end{aligned}$$

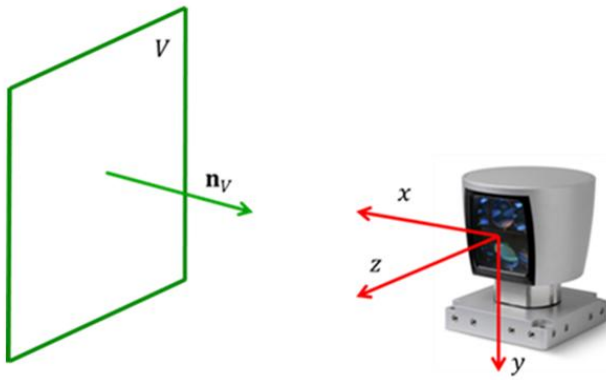

Figure 3.5: ASPN LiDAR Frame

3.11.9 3D Laser Scanners. 3D Laser scanners as used in this research provide geometric point clouds of the environment around them. This is done using pulses of light with a measured time of arrival at a known angle of the pulsed. In navigation, 3D laser scanners measurements can be used to provide both positioning and attitude information.

3.11.9.1 Preprocessed Measurements. For this research, 3D Laser measurements are provided as planar surfaces and processed to include motion compensation, planar surface feature extraction, and correspondence between subsequent planar features.

The planar surface measurement is modeled as:

$$\boldsymbol{\zeta}_{meas} = \begin{bmatrix} \mathbf{n}_{true} \\ d_{true} \end{bmatrix} + \mathbf{w}_{L3} \tag{3.171}$$

where $\boldsymbol{\zeta}$ is the planar surface extracted from the point cloud, $\mathbf{w}_{L3}$ is the measurement white Gaussian noise error, $\mathbf{n}$ is the plane normal with components:

$$\mathbf{n} = \begin{bmatrix} a & b & c \end{bmatrix}^T$$

which together with d correspond to the parameters of the definition of a plane as:

$$ax + by + cz + d = 0 \tag{3.172}$$

The white Gaussian noise covariance is provided through the point cloud preprocessing and represented by:

$$\mathbf{R}_{L3} = E[(\mathbf{w}_{L3})(\mathbf{w}_{L3})^T] = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} & \sigma_{ac} & \sigma_{ad} \\ \sigma_{ba} & \sigma_b^2 & \sigma_{bc} & \sigma_b \\ \sigma_{ca} & \sigma_{cb} & \sigma_c^2 & \sigma_{cd} \\ \sigma_{da} & \sigma_{dc} & \sigma_{dc} & \sigma_d^2 \end{bmatrix} \tag{3.173}$$

3.11.9.2 Position Measurement Update. The 3D laser scanner native measurement equation provides the defining parameters of planar surfaces resolved in the laser scanner frame. Positioning information is gleaned from the measurements by calculating the shortest distance ρ from the laser scanner to the plane. This is given by the projection of the plane normal onto any vector pointing from the plane to the laser scanner given from [27] as:

$$\rho = \frac{|d|}{\|\mathbf{n}\|} \tag{3.174}$$

This can be simplified by removing the absolute value operator and changing the sign on d to be negative, since the laser scanner is at the frame origin which will always be opposite to the plane normal. The closest distance is then given by:

$$\rho = \frac{-d}{\|\mathbf{n}\|} \tag{3.175}$$

The closest distance is further simplified if we assume the plane normal is a unit vector which is done from here on giving:

$$\rho = -d \tag{3.176}$$

and $\mathbf{n}$ will henceforth be written in the unit vector notation as $\bar{\mathbf{n}}$. The change in the closest distance to the same plane seen in a subsequent scan gives the change in position in the direction of the plane normal resolved in the previous laser scanner frame. From [26], this is expressed as:

$$\rho_{k-1} - \rho_k = \Delta\mathbf{p}^s \cdot \bar{\mathbf{n}}^i_{k-1} \tag{3.177}$$

where $\Delta\mathbf{p}^s$ is the position change in the laser scanner frame since the last laser scanner measurement, $\bar{\mathbf{n}}$ is to the plane normal, and i and j are the previous and current scanner reference frames respectively.

The position measurement equation is seen to be related directly to the position states, by first performing a rotation from the laser scanner frame to the body frame and then using the current best attitude estimate to rotate from the body to NED. This is given as:

$$\mathbf{z}_{L3_{pos}} = (\mathbf{C}^s_b \mathbf{C}^b_n \Delta\mathbf{p}^n) \cdot \bar{\mathbf{n}}^i_{k-1} \tag{3.178}$$

where $v_{L3_{pos}}$ is white Gaussian noise measurement error.

Similar to the position update, the change in the plane normals in the laser scanner frame provides information about the attitude change. From [26] the relationship between $\bar{\mathbf{n}}^i_{k-1}$ and $\bar{\mathbf{n}}^j_k$ is given by the rotation:

$$\bar{\mathbf{n}}^j_k = \mathbf{C}^j_i \bar{\mathbf{n}}^i_{k-1} \tag{3.179}$$

where $\mathbf{C}^j_i$ is the DCM representing the rotation between the two scan times and can be expressed directly in terms of the quaternion states and the laser scanner to body DCM. This is accomplished through several steps. First the attitude change must be calculated since the last scan was taken. This is a function of the previous and current quaternion attitude and is calculated from (2.23) as:

$$\Delta\mathbf{q} = \mathbf{Q}(\mathbf{q}_k)\mathbf{q}^*_{k-1} \tag{3.180}$$

Populating (2.15) with the elements of $\Delta\mathbf{q}$, the attitude change DCM from j to i ($\mathbf{C}_j^i$) is formed.

3.11.9.3 Complete Measurement Update. The complete measurement update is then built by vertically concatenating corresponding components of the measurement update equations for both position and attitude. The measurement noise covariance is calculated as:

$$\mathbf{z}_{L3} = \mathbf{h}_{L3} + \mathbf{w}_{L3} \tag{3.181}$$

$$\mathbf{z}_{L3} = \begin{bmatrix} \bar{\mathbf{n}}_{meas_k} \\ d_{meas_k} - d_{meas_{k-1}} \end{bmatrix} \tag{3.182}$$

$$\mathbf{h}_{L3} = \begin{bmatrix} \mathbf{C}_i^j \bar{\mathbf{n}}_{k-1}^i \\ (\mathbf{C}_b^s \mathbf{C}_n^b (\mathbf{p}_k^n - \mathbf{p}_{k-1}^n)) \cdot \bar{\mathbf{n}}_{k-1}^i \end{bmatrix} \tag{3.183}$$

$$\mathbf{H}_{L3} = \left. \frac{\partial \mathbf{h}_{L3}}{\partial \mathbf{x_k}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k-1}} \tag{3.184}$$

$$\mathbf{J}_{L3} = \left. \frac{\partial \mathbf{h}_{L3}}{\partial \mathbf{x_{k-1}}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k-1}} \tag{3.185}$$

$$\mathbf{R}_{L3} = \begin{bmatrix} & & & \sigma_{ad_k} \\ & \mathbf{R_n} & & \sigma_{b_k} \\ & & & \sigma_{cd_k} \\ \sigma_{da_k} & \sigma_{dc_k} & \sigma_{dc_k} & \sigma_{d_k}^2 + \sigma_{d_{k-1}}^2 \end{bmatrix} \tag{3.186}$$

where

$$\mathbf{R_n} = \begin{bmatrix} \sigma_{a_k}^2 & \sigma_{ab_k} & \sigma_{ac_k} \\ \sigma_{ba_k} & \sigma_{b_k}^2 & \sigma_{bc_k} \\ \sigma_{ca_k} & \sigma_{cb_k} & \sigma_{c_k}^2 \end{bmatrix} + \mathbf{C}_i^j \begin{bmatrix} \sigma_{a_{k-1}}^2 & \sigma_{ab_{k-1}} & \sigma_{ac_{k-1}} \\ \sigma_{ba_{k-1}} & \sigma_{b_{k-1}}^2 & \sigma_{bc_{k-1}} \\ \sigma_{ca_{k-1}} & \sigma_{cb_{k-1}} & \sigma_{c_{k-1}}^2 \end{bmatrix} \mathbf{C}_j^i \tag{3.187}$$

3.11.10 3-Axis Magnetometer. The 3-Axis magnetometer provides possible position updates based on the earth's magnetic field mapped to specific geodetic locations. The mappings are generated from previous passes along the same route

referencing standalone GPS positioning data. On subsequent passes when GPS is not available or degraded, the 3-Axis magnetometer provides high likelihood values of the current position based off of past data.

3.11.10.1 Measurement Update. The measurement update is based off the current estimated position compared to the high likelihood values provided by the sensor. In this thesis, the measurement is given by selecting the maximum high likelihood position update. The measurement is modeled as:

$$\begin{bmatrix} L_{meas} \\ \lambda_{meas} \\ Alt_{meas} \end{bmatrix} = \begin{bmatrix} L_{true} \\ \lambda_{true} \\ Alt_{true} \end{bmatrix} + \mathbf{w}_{magn} \tag{3.188}$$

where L is the geodetic latitude, λ is longitude, Alt is the altitude, and $\mathbf{w}_{magn}$ is given by:

$$\mathbf{w}_{magn} = \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}_{magn} \tag{3.189}$$

where R_E is the radius of the earth, and $\boldsymbol{\nu}_{magn}$ is the white Gaussian noise measurement position error in meters. The magnetometer lever arm gives a spatial displacement to the position measurement as:

$$\mathbf{p}^n_{magn} = \begin{bmatrix} L \\ \lambda \\ Alt \end{bmatrix} + \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}^g_n \mathbf{C}^n_b \mathbf{l}^b \tag{3.190}$$

where $\mathbf{l}^b$ is the sensor lever arm. The measurement update equation is the incorporation of (3.212) with respect to the estimated geographic states as:

$$\mathbf{z}_{magn} = \mathbf{h}_{magn} + \mathbf{w}_{magn} \tag{3.191}$$

$$\mathbf{z}_{magn} = \begin{bmatrix} L_{meas} & \lambda_{meas} & Alt_{meas} \end{bmatrix}^T \tag{3.192}$$

$$\mathbf{h}_{magn} = \mathbf{p}^n_{magn} \tag{3.193}$$

$$\mathbf{H}_{magn} = \Big[\underbrace{\mathbf{I}_3}_{L\lambda Alt} \Big| \; \mathbf{0}_3 \quad \cdots \quad \mathbf{0}_3 \Big]_{3\times N} \tag{3.194}$$

$$\mathbf{R}_{magn} = E[(\mathbf{w}_{magn})(\mathbf{w}_{magn})^T] \tag{3.195}$$

3.11.11 Odometry Sensors. Odometers provide measurements of relative position change. This represents total distance traveled without regard to attitude or direction of travel. In this project, the odometer measurement has no memory beyond each time epoch and thus reports the distance traveled over the past time step. The odometer measurment differs slightly from the traditional odometer in that it also provides negative distances when the platform moves in reverse. As such, the native measurement presented by the odometer is given as:

$$\Delta D_{meas} = \Delta D_{true}(1 + SF) + w_{odom} \tag{3.196}$$

Where ΔD represents the change in total distance traveled, SF is a constant scale-factor bias and w_{odom} is the measurement white Gaussian noise error.

3.11.11.1 Measurement Update. The measurement is then related to the states by taking the magnitude of change in the NED frame. This can be computed directly from the change in latitude, longitude, and altitude states from (2.14) using the previous and current position estimate. Including the effects of the lever arm, which is the distance from the origin of the body frame to the odometer

path contact surface, the change in position is given as:

$$\Delta\mathbf{r}^n = \begin{bmatrix} (R_p + Alt_k)(L_k - L_{k-1}) \\ (R_m + Alt_k)\cos L_k(\lambda_k - \lambda_{k-1}) \\ Alt_{k-1} - Alt_k \end{bmatrix} + \mathbf{C}_b^n(\psi_k - \psi_{k-1}) \begin{bmatrix} 0 \\ l_y^b \\ 0 \end{bmatrix} \tag{3.197}$$

where ψ is the heading angle and l_y^b is the horizontal component of the lever arm in the body frame. Following the delayed-state measurement update form as outlined in Section 2.9.3, the final delayed-state stochastic measurement equations are then expressed as:

$$\begin{aligned} \mathbf{z}_{odom} &= h_{odom} + w_{odom} && (3.198) \\ z_{odom} &= \Delta D_{meas} && (3.199) \\ h_{odom} &= \|\Delta\mathbf{r}^n\|(1 + SF) && (3.200) \\ \mathbf{H}_{odom} &= \left.\frac{\partial \mathbf{h}_{odom}}{\partial \mathbf{x_k}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k-1}} && (3.201) \\ \mathbf{J}_{odom} &= \left.\frac{\partial \mathbf{h}_{odom}}{\partial \mathbf{x_{k-1}}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k-1}} && (3.202) \\ \mathbf{R}_{odom} &= E[(w_{odom})(w_{odom})^T] && (3.203) \end{aligned}$$

3.11.12 Ranging Sensors. The ranging sensor measures the distance to a beacon of known WGS-84 position. This provides a measurement similar to the GNSS pseudorange measurement. The measurement is modeled as:

$$r_{meas} = r_{true} + b_{range} + w_{range} \tag{3.204}$$

where r is the range to the beacon, b_{range} is a TCB that models the sensor induced bias over time, and w_{range} is the white Gaussian noise measurement hardware error. The range to the beacon is then expressed in terms of the states and beacon coordinates

by forming the distance vector in the NED frame as:

$$\mathbf{r}^n = \begin{bmatrix} (R_p + Alt_b)(L_b - L) \\ (R_m + Alt_b)\cos L_b(\lambda_b - \lambda) \\ Alt - Alt_b \end{bmatrix} + \mathbf{C}_b^n \mathbf{l}^b \tag{3.205}$$

where L_b, λ_b, and Alt_b are the WGS-84 coordinates of the beacon and $\mathbf{l}^b$ is the ranging sensor lever arm. The range to the beacon is given in terms of the states as:

$$D = \|\mathbf{r}^n\| \tag{3.206}$$

The filter measurement update equations are then presented as:

$$z_{range} = h_{range} + w_{range} \tag{3.207}$$

$$z_{range} = r_{meas} \tag{3.208}$$

$$h_{range} = \|\mathbf{r}^n\| + b_{range} \tag{3.209}$$

$$\mathbf{H}_{range} = \left.\frac{\partial h_{range}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.210}$$

$$\mathbf{R}_{range} = E[(w_{range})(w_{range})^T] \tag{3.211}$$

3.11.13 Radio Frequency Identification Sensors. RFID devices continue to increase in popularity for a multitude of reasons. From replacing the familiar barcodes on products to tracking pets, RFID devices are quickly working their way into everyday life. Consisting of a silicon microchip and an antennae, RFID tags are now found on the sub-millimeter level and enable identification from a distance without requiring a line of sight [33]. Their small size and dramatically decreasing cost make them an alluring option for a multitude of applications including tracking and navigation.

Many types of RFID exist however, they are generally divided into two classes of active or passive operation. Passive RFID tags operate only off the signal power

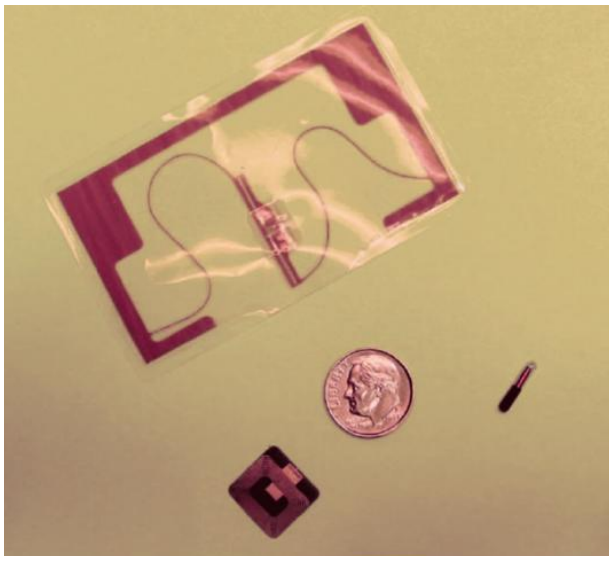

Figure 3.6: From [33], three different RFID tags and their relative size with an everyday object.

of an RFID tag reader. Possessing no power source allows the tag to be smaller and cheaper, but has the disadvantage of very limited range. Active RFID tags possess their own power source and are typically larger and more expensive than passive models. However their signal range is significantly larger than their passive counterparts. Though they can be matched with alternate equipment (e.g. memory chip, GNSS receiver, etc.) the RFID tags considered in this project simply transmit their identification code when queried. This provides a unique identifying code to a reader. Tracking can be performed by knowing either the location of the reader or RFID tag depending on which has motion and which is stationary. For tracking purposes in an environment with many tags, a passive RFID tag is more effective for pinpointing location. The opposite is true for environments with sparse tags.

For this project, stationary RFID tags are used with known WGS-84 coordinates. A database is generated that contains the position, position uncertainty, and operation type matched to each RFID tag ID. Both passive and active RFID tags are used in this project.

The RFID sensor measurement is given by the matching the tag ID with the database location an modeled as:

$$\begin{bmatrix} L_{meas} \\ \lambda_{meas} \\ Alt_{meas} \end{bmatrix} = \begin{bmatrix} L_{true} \\ \lambda_{true} \\ Alt_{true} \end{bmatrix} + \mathbf{w}_{RFID} \tag{3.212}$$

where L is the geodetic latitude, λ is longitude, Alt is the altitude, and $\mathbf{w}_{RFID}$ is given by:

$$\mathbf{w}_{RFID} = \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu}_{RFID} \tag{3.213}$$

where R_E is the radius of the earth, and $\boldsymbol{\nu}_{RFID}$ is the white Gaussian noise measurement position error in meters. The covariance of $\boldsymbol{\nu}_{RFID}$ is a combination of the range that the tag can be identified and the uncertainty in the tag geodetic location which is known ahead of time. It is expressed as:

$$\sigma^2_{RFID} = (\frac{1}{2} D_{RFID})^2 + \sigma^2_{RFID_{\mathbf{p}^n}} \tag{3.214}$$

where D_{RFID} is the max range the tag can be identified, and $\sigma_{RFID_{\mathbf{p}^n}}$ is the tag geodetic location standard deviation expressed in meters. The RFID device lever arm is included in the measurement equation as:

$$\mathbf{p}^n_{RFID} = \begin{bmatrix} L \\ \lambda \\ Alt \end{bmatrix} + \begin{bmatrix} \frac{1}{R_E} & 0 & 0 \\ 0 & \frac{1}{R_E \cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}^g_n \mathbf{C}^n_b \mathbf{l}^b \tag{3.215}$$

where $\mathbf{l}^b$ is the sensor lever arm. This acts to place the RFID reader at the location of the RFID tag rather than the platform navigation system location. This is only performed with passive tags where it is logical to assume that the RFID reader and tag lie on the same side of the platform. Otherwise the lever arm is ignored. The

measurement update equations presented to the filter are given as:

$$\mathbf{z}_{RFID} = \mathbf{h}_{RFID} + \mathbf{w}_{RFID} \tag{3.216}$$

$$\mathbf{z}_{RFID} = \begin{bmatrix} L_{meas} & \lambda_{meas} & Alt_{meas} \end{bmatrix}^T \tag{3.217}$$

$$\mathbf{h}_{RFID} = \mathbf{p}^n_{RFID} \tag{3.218}$$

$$\mathbf{H}_{RFID} = \Big[\underbrace{\mathbf{I}_3}_{\mathbf{p}^n} \Big| \; \mathbf{0}_3 \quad \cdots \quad \mathbf{0}_3 \Big]_{3\times N} \tag{3.219}$$

$$\mathbf{R}_{RFID} = E[(\mathbf{w}_{RFID})(\mathbf{w}_{RFID})^T] \tag{3.220}$$

3.11.14 Signal of Opportunity Sensors. Signal of opportunity sensors are simulated sensors that measure the time difference of arrival from two time synchronized beacons. The measurement is the difference in signal propagation time from the source to the mobile platform and the reference beacon. The native measurement equation is modeled as:

$$TDOA_{meas} = TDOA_{true} + b_{TDOA} + w_{TDOA} \tag{3.221}$$

where $TDOA$ is the time difference of delay, b_{TDOA} is a TCB bias that represents the combined clock error of the two signals, and w_{TDOA} is white Gaussian noise measurement error.

3.11.14.1 Measurement Update. From Figure 3.7, the distance from the source to the mobile platform, D_{mp}, and the distance from the source to the reference beacon, D_{ref}, are both related through the time difference of delay by:

$$TDOA = \frac{D_{mp}}{c} - \frac{D_{Ref}}{c} + b_{TDOA} \tag{3.222}$$

where c is the speed of light and the signal propagation speed. The distance from the source to the reference beacon D_{Ref} is known and its uncertainty is denoted by σ^2_{Ref}. Rearranging (3.222), such that the propagation time is associated with the TDOA,

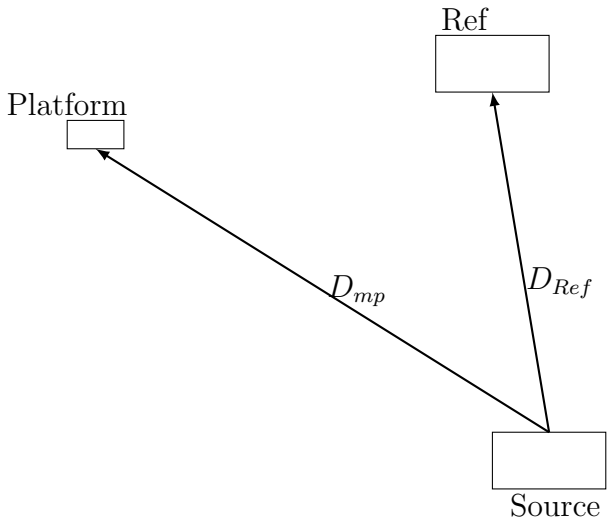

Figure 3.7: Signal of opportunity measurement setup.

and the distance from the source to the platform is related to the position states as:

$$D_{mp} = \|\mathbf{r}^n\|,\ \mathbf{r}^n = \begin{bmatrix} (R_p + Alt)(L - L_{source}) \\ (R_m + Alt)\cos L(\lambda - \lambda_{source}) \\ Alt_{source} - Alt \end{bmatrix} + \mathbf{C}_b^n \mathbf{l}^b \tag{3.223}$$

where $\mathbf{l}^b$ is the sensor lever arm in the body frame. Though the sensor measurement is in units of time, it is numerically beneficial to convert the measurement into meters. This reduces burden on the filter since the signal propagates at the sped of light, and using very small numbers with very large numbers can quickly lead to ill conditioning in the filter. The complete set of measurement update equations are then given as:

$$z_{SOS} = h_{SOS} + w_{TDOA} \tag{3.224}$$

$$z_{SOS} = c(TDOA) + D_{Ref} \tag{3.225}$$

$$h_{SOS} = D_{mp} + b_{TDOA_m} \tag{3.226}$$

$$\mathbf{H}_{SOS} = \left.\frac{\partial h_{SOS}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{3.227}$$

$$\mathbf{R}_{SOS} = c^2 E[(w_{TDOA})(w_{TDOA})^T] + \sigma_{ref}^2 \tag{3.228}$$

where σ_{ref} is the standard deviation of the reference position in meters, b_{TDOA_m} is the clock error bias now estimated in meters. The variance and process noise strength on b_{TDOA_m} must be appropriately modified to account for the change of units.

3.11.15 Step Sensor. The step sensor provides information that can be used to estimate relative position change. This represents total distance traveled without regard to attitude or direction of travel. The step sensors indicates when a step has taken place. As such, the measurement is modeled as:

$$\Delta D_{meas} = \Delta D_{true} + b_{step} + w_{step} \tag{3.229}$$

Where ΔD represents the change in total distance traveled, b_{step} is a TCB, and w_{step} is the measurement white Gaussian noise error. Since no tangible measurement is actually presented, the measurement is set prior to filter execution to an average human step size. The bias term in (3.229) then accounts for differences in individual gait and walking speed which would also affect the distance between steps.

3.11.15.1 Measurement Update. The measurement is then related to the states by taking the magnitude of change in the NED frame. This can be computed directly from the change in latitude, longitude, and altitude states from (2.14) using the previous and current position estimate as:

$$\Delta \mathbf{r}^n = \begin{bmatrix} (R_p + Alt_k)(L_k - L_{k-1}) \\ (R_m + Alt_k)\cos L_k(\lambda_k - \lambda_{k-1}) \\ Alt_{k-1} - Alt_k \end{bmatrix} \tag{3.230}$$

Following the delayed-state measurement update form as outlined in Section 2.9.3, the final delayed-state stochastic measurement update equations are then expressed

as:

$$
\begin{aligned}
z_{step} &= h_{step} + w_{step} && (3.231)\\
z_{step} &= \Delta D_{meas} && (3.232)\\
h_{step} &= \|\Delta \mathbf{r}^n\| + b_{step} && (3.233)\\
\mathbf{H}_{step} &= \left.\frac{\partial \mathbf{h}_{step}}{\partial \mathbf{x_k}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_{k-1}} && (3.234)\\
\mathbf{J}_{step} &= \left.\frac{\partial \mathbf{h}_{step}}{\partial \mathbf{x_{k-1}}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_{k-1}} && (3.235)\\
\mathbf{R}_{step} &= E[(w_{step})(w_{step})^T] && (3.236)
\end{aligned}
$$

3.11.16 Stop Sign Sensor. The stop sign sensor is realized by some optical sensor mounted on the platform capable of detecting stop signs. For this project there is no orientation for the sensor. The only information given by the sensor is that a stop sign has been spotted. A database of all known stop sign geodetic latitudes and longitudes along with the heading angle of the normal vector to the stop sign face is must be generated.

3.11.16.1 Modeling. The stop sign sensor region of detection is portrayed in Figure 3.8. Once a sign is detected, the stop sign database is searched for the top ten closest stop signs to the current estimated position. For a single stop sign, the mean position update is determined in the center of the stop sign region of detection. The distance from the stop sign given by:

$$d = \frac{r_2 - r_1}{2} + r_1 \tag{3.237}$$

where r_1 is the minimum distance that the sensor must be from the stop sign and r_2 is the maximum distance that the sensor can be from the stop sign. The uncertainty

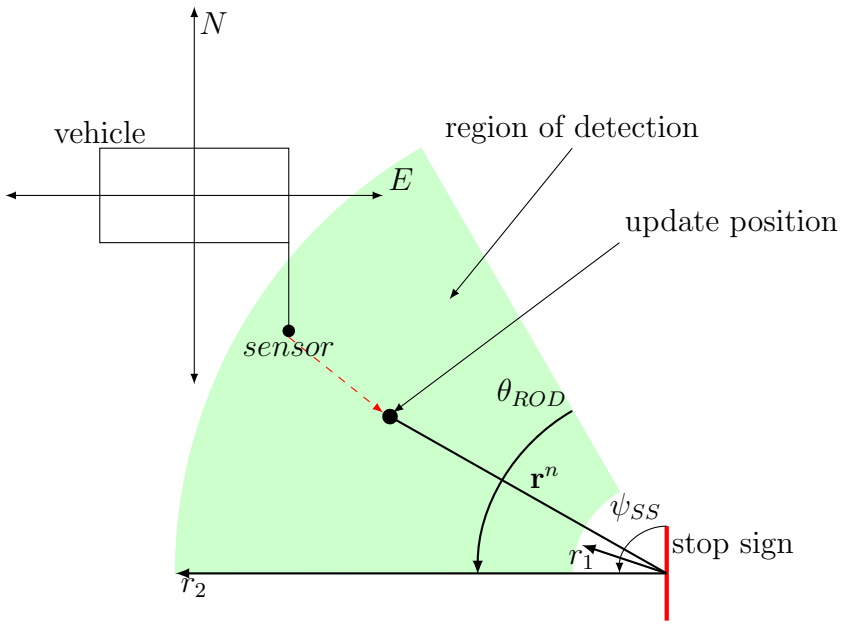

Figure 3.8: Stop sign sensor region of detection.

of the stop sign sensor in the region of detection is given by:

$$\mathbf{R}_d = \begin{bmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_{\psi_{SS}}^2 \end{bmatrix}, \sigma_d = \frac{r_2 - r_1}{2} \text{ and } \sigma_{\psi_{SS}} = \theta_{ROD} \tag{3.238}$$

where θ_{ROD} is the region of detection angle. The two dimensional vector (stop sign altitude is unknown) extending from the stop sign to the update position is given by:

$$\mathbf{d} = \mathbf{C}_{SS} \begin{bmatrix} d \\ 0 \end{bmatrix} \tag{3.239}$$

where L_{SS} and λ_{SS} represent the geodetic location of the sign stop given in the database, and $\mathbf{C}_{SS}$ is given by:

$$\mathbf{C}_{SS} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & cos\theta \end{bmatrix}, \ \theta = \frac{\theta_{ROD}}{2} + \psi_{SS} \tag{3.240}$$

where ψ_{SS} is the stop sign heading angle. The update position for the vehicle in the NED frame is then given by placing the sensor at the specified distance, d, and

orientation, θ, from the sign and converting the position to the geodetic navigation frame as:

$$\begin{bmatrix} L_{meas} \\ \lambda_{meas} \end{bmatrix} = \begin{bmatrix} L_{SS} \\ \lambda_{SS} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_E} & 0 \\ 0 & \frac{1}{R_E \cos L_{SS}} \end{bmatrix} \mathbf{d} + \mathbf{w}_{SS} \tag{3.241}$$

where $\mathbf{w}_{SS}$ is the white Gaussian noise process that characterizes the uncertainty in the stop sign position and the measurement dependence on the current position. The covariance of $\mathbf{w}_{SS}$ is given as:

$$E[(\mathbf{w}_{SS})(\mathbf{w}_{SS})^T] = \mathbf{P}_{\mathbf{x}_{L\lambda}} + \mathbf{P}_{SS_{L\lambda}} + \begin{bmatrix} \frac{1}{R_E^2} & 0 \\ 0 & \frac{1}{(R_E \cos L)^2} \end{bmatrix} \mathbf{W} \begin{bmatrix} d \\ \theta \end{bmatrix} \mathbf{W}^T \tag{3.242}$$

where $\mathbf{P}_{\mathbf{x}_{L\lambda}}$ is the filter provided covariance of the latitude and longitude states, $\mathbf{P}_{SS_{L\lambda}}$ is the database provided covariance of the latitude and longitude of the stop sign, and $\mathbf{W}$ is the influence matrix given by:

$$\mathbf{W} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ -\sin\theta & -d\cos\theta \end{bmatrix} \tag{3.243}$$

For consistency with other update equations in this thesis, the lever arm offset is associated with the states in the measurement update equations. The first two elements of the lever arm, body to NED frame DCM multiplication,$\mathbf{C}_b^n\mathbf{l}^b$, gives the lever arm offset $\mathbf{l}_{NE}$. The final measurement update equations are then given as:

$$\mathbf{z}_{SS} = \mathbf{h}_{SS} + \mathbf{w}_{SS} \tag{3.244}$$

$$\mathbf{z}_{SS} = \begin{bmatrix} L_{meas} & \lambda_{meas} \end{bmatrix}^T \tag{3.245}$$

$$\mathbf{h}_{SS} = \begin{bmatrix} L & \lambda \end{bmatrix}^T + \begin{bmatrix} \frac{1}{R_E^2} & 0 \\ 0 & \frac{1}{(R_E \cos L)^2} \end{bmatrix} \mathbf{l}_{NE} \tag{3.246}$$

$$\mathbf{H}_{SS} = \Big[\underbrace{\mathbf{I}_2}_{L\lambda} \Big| \ \mathbf{0} \Big]_{2\times N} \tag{3.247}$$

$$\mathbf{R}_{SS} = E[(\mathbf{w}_{SS})(\mathbf{w}_{SS})^T] \tag{3.248}$$

To determine the stop sign that was observed, an array of pseudo measurements is formed using each stop sign in consideration. A trial of incorporating each stop sign position update into the filter is performed. The stop sign measurement update that minimizes the measurement residual given in (2.57) and stays within 3σ of the residual covariance is selected as the final filter measurement update. If no stop sign position measurement update meets the 3σ criteria, then no stop sign sensor measurement is used. The stop sign sensor measurement update and the position measurement update are highly correlated. As a result the stop sign sensor is expected to sometimes provide incorrect measurements causing significant errors in the navigation solution.

3.11.17 Terrain-Referenced Altimeter. The terrain-referenced altimeter (TRA) measures the distance from the sensor to the ground plane. Its measurement is modeled as:

$$D_{meas} = D_{true} + b_D + w_D \tag{3.249}$$

where D is the distance from the sensor to the ground plane, b_D is a TCB bias due to the sensor, and w_D represents the white Gaussian noise measurement error.

3.11.17.1 Measurement Update. To relate the measurement to the states, information regarding the terrain height is required. This is provided by a digital terrain elevation data (DTED) map that provides a terrain height estimate given a specific latitude and longitude. Associating the DTED map information with the measurement rather than the estimate, the measurement equation is related directly to the states as:

$$D_{meas} + Alt_{DTED} = Alt + b_D + w_{TRA} \tag{3.250}$$

where Alt_{DTED} is the terrain height above MSL, and w_{TRA} is the combined uncertainty in the measurement and the DTED terrain height given as:

$$w_{TRA} = w_D + w_{DTED} \tag{3.251}$$

This measurement is obviously correlated with with latitude and longitude position states. Because it is a measurement, this relationship cannot be shown directly in the filter state covariance measurement using the traditional EKF, but it must at least be accounted for in w_{DTED}.

The height from DTED is not taken directly from the dataset, but estimated by sampling the terrain height in the region of latitude and longitude uncertainty and calculating a terrain height standard deviation and mean. This is accomplished using the UT where sigma points are generated to reflect a mean latitude and longitude with given uncertainty. DTED provides the transformation function and the mean and covariance of the sigma points provide mean DTED height and uncertainty.

The final measurement update equations are then presented to the filter as:

$$z_{TRA} = h_{TRA} + w_{TRA} \tag{3.252}$$

$$z_{TRA} = D_{meas} + Alt_{DTED} \tag{3.253}$$

$$h_{TRA} = Alt + b_D \tag{3.254}$$

$$\mathbf{H}_{TRA} = \Big[\underbrace{0 \quad 0 \quad 1}_{L\lambda Alt} \Big| \; 0 \quad \dots \quad 0 \; \Big| \underbrace{1}_{b_D} \Big| \; 0 \quad \dots \quad 0 \; \Big]_{1\times N} \tag{3.255}$$

$$R_{TRA} = E[(w_{TRA})(w_{TRA})^T] \tag{3.256}$$

This measurement is highly correlated with the latitude and longitude positions states. The relationship cannot be expressed using the traditional EKF measurement update equations and can lead to suboptimal performance in the navigation solution.

3.12 Summary

Chapter III outlined the steps taken to develop the EKF filter used in this research. A measurement incorporation, method was adopted to treat all sensor provided information in the traditional Kalman filter, measurement update sense. Basic navigational states where chosen to fit the needs of each of the three platforms

and the adopted update method. A generic dynamics model to was developed that could be tuned to meet the requirements of each platform. The dynamics model shows the relationship between all the estimated states, and it was based upon a first-order Gauss-Markov acceleration and angular rate model.

For each sensor, a measurement model was defined which described the measurements and their contained error quantities. For some sensors, error states had to be estimated to allow effective use of their measurements. Sensor specific states were described as well as their incorporation into the state estimate vector and state covariance matrix. The measurements were then related to the estimated states and the associated EKF update equations were given for each sensor.

To accommodate the specific needs of each sensor, a flexible filter was developed to adjust the state estimate vector and the state covariance matrix. The filter operates in a manner that could be implemented in real time. Measurements are sensed in the order that they were taken, and the filter propagates to the measurement time of validity immediately prior to incorporating the measurement.

In the next chapter, scenarios that the filter was tested on are discussed and results are presented. This includes simulation data generation, sensor characteristics, sensor set descriptions, and navigation tracking performance under various sensor configurations.

IV. Results

The results and simulations developed in this thesis are discussed in this chapter. This chapter is organized first by discussing the paths simulation software, sensor measurement simulations, sensor characteristics, and the tracking scenarios. Sensors are added in simulations to demonstrate their contribution to the navigation solution. Navigation tracking results are provided in terms of the estimated state error and its associated standard deviation (STD).

4.1 Simulations

Simulations where designed to test the utility of each additional sensor to the sensor set. Each simulation was conducted in three dimensional space, although several have one or more axes constrained. Some sensor measurements provide overpowering state observability when compared other sensors. To demonstrate the performance of the subtler sensor measurements, simple straight paths were created for both the vehicle and the pedestrian.

4.1.1 Trajectory Simulation Software. All simulation platform trajectories were computed using Profgen Tools. Profgen Tools consists of four computer programs for computing the position and attitude of platform given a list of motion commands. The motion commands include movements, such as the forward and backward accelerations, vertical and horizontal turns, roll maneuvers, jinking, and free fall. The software is developed around a model for an aircraft, but has been used for other platforms including public transit buses. Profgen is limited to single aircraft at a time modeled as a point mass. This allows kinematic data to be created easily while minimizing platform characteristics requirements such as mass, weight, thrust, etc. [22].

The Profgen code provides all the truth data for basic navigation states discussed in Chapter III, as well as modeled and known quantities such as gravity and earth sidereal rate. For most sensors, this information is sufficient for a complete

simulation, however other sensors such as cameras and 3D laser scanners require additional information to use the measurement models presented in Chapter III. Sensors that required a modest amount of additional coding to simulate are described next.

4.1.2 RFID Tags. To simulate randomly placed RFID tags in each scenario, an autonomous routine was built for the RFID sensors that takes the total distance traveled in a scenario, divides the total distance by a predetermined segment length, and randomly distributes a predetermined number of RFID tags. The distribution of the tags placement is Gaussian with a predetermined standard deviation displacing them from the traveled path. The standard deviation for the horizontal placement is considerably great than the standard deviation for the vertical placement to avoid tags being placed very far below or above the platform. Each RFID tag by default is made passive with a few active RFID tags placed manually throughout each track.

Once created, the tags locations are saved so that the same set can be considered in multiple sensor set configurations. Measurements are developed in the simulator by moving the platform along the simulated path and calculating the distance from the sensor to each tag at each time step. The step metric is controlled by a user defined sample time. Measurements are registered when the range falls below the threshold set by the passive or active tag range specification.

4.1.3 Camera Features. Placement of the camera features follows the same procedure are with the RFID tags with the exception that very many features are usually placed at much shorter segment lengths as seen in Figure 4.1. Once the features have been placed, the platform is moved along the simulated path. At each time step a range and angle from the camera is calculated as:

$$r = \|\mathbf{s}^c\| \tag{4.1}$$

$$\theta = \cos^{-1}(\mathbf{s}^c_z / r) \tag{4.2}$$

where $\mathbf{s}^c$ is the projection vector from the camera to the feature and s^c_z is the z

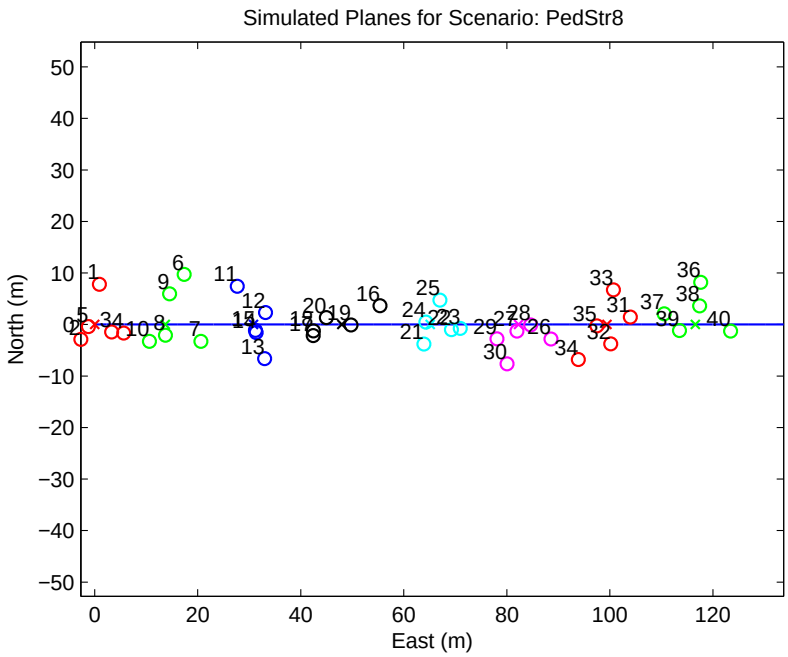

Figure 4.1: Example of random camera feature generation

component of the projection vector. Features that are within a predetermined minimum and max range as well as a representative field of view are then registered as measurements. This makes the camera measurement more representative of physical hardware while ensuring that measurements are never taken for features that are behind the camera (i.e. where $s_z^c < 0$). This would violate a fundamental assumption of the camera projection model and cause problems as there is no way to tell from the homogenous measurement vector that the feature was actually behind the camera. The same process is performed with stereo cameras with the additional stipulation of meeting the requirements of the range and field of view imposed by both cameras. Once measurements are registered, error is simulated using white Gaussian noise as shown in the camera measurement model.

4.1.4 Laser 3D Planes. Generation of planes that would be extracted from 3D point clouds in a real laser scanner follow the same random path procedure as with the RFID tags and the camera features. There are more planes randomly placed at each segment than RFID tags and less than camera features. At each segment of

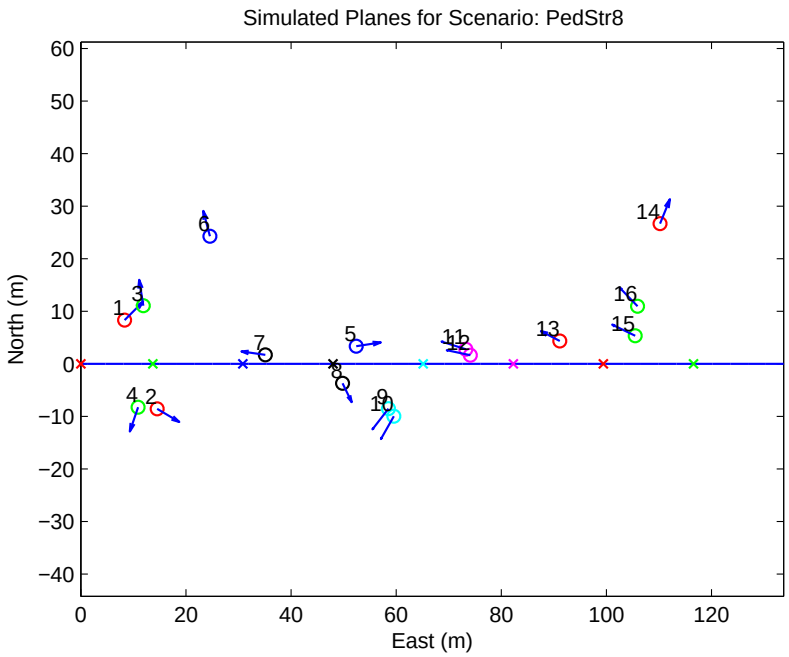

Figure 4.2: Top view of uncorrected 3D scanner plane normals at the positions in the NED frame where the planes are defined.

the track, planes are created at a point where the line connecting the segment and point becomes the plane normal vector. This point is chosen to represent the physical location of the plane in space although the planes are modeled as infinite in size. This is required to simulate the range limit of a real sensor where the plane is derived from some flat surface that has a physical interpretation such as a wall. Even with real data the planes are modeled as to have infinite size for this implementation. Points are chosen about the segment so that plane normals are mostly horizontal to reflect many real life urban scenarios. For this project, the plane normal is always defined in the NED frame pointing away from the origin. That is, the side of the plane that the origin lies is opposite to the side the plane normal points from. Because of this, planes with normals that point towards the origin, as seen in Figure 4.2, must be corrected. This is done by checking the distance to the plane form the origin using (3.175). If the origin lies on the same side of the plane that the normal points, the distance will be negative [34]. Performing a negative distance check, planes that fail the test have their normals reversed and the fourth defining parameter of the plane

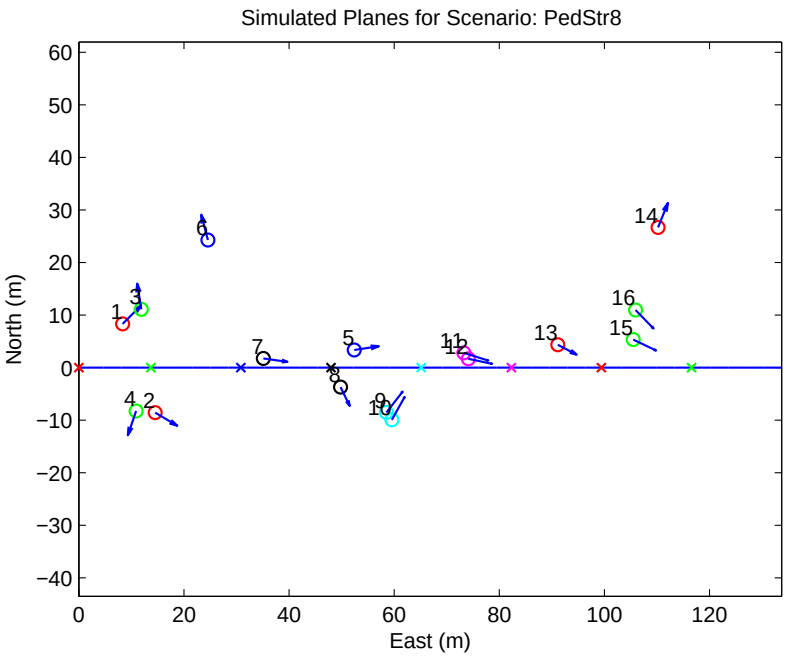

Figure 4.3: Top view of the corrected 3D scanner plane normals at the positions in the NED frame where the planes are defined.

is reevaluated. The complete correction is given by:

$$\boldsymbol{\zeta}_{new} = -\boldsymbol{\zeta}_{old} \tag{4.3}$$

where $\boldsymbol{\zeta}$ represents the four parameters that define a plane described in Section 3.11.9 and the corrected plane normals are shown in Figure 4.3.

Once the set of planes has been created, the measurement simulator calculates a range to the point that each plane was previously defined to originate from. This is performed at each time step, which is again user defined. All planes that are in range are registered as measurements, and measurement error is added using white Gaussian noise.

4.1.5 Sensor Set. To generate the scenarios, a large database of sensors was compiled. The sensors that composed each scenario's sensor sets, were selected from the main database of sensors. The database contained all pertinent metadata for each sensor. Separate configuration files for each scenario were written that included information such as sensor orientation and the lever arm. The simulated scenarios described in Sections 4.2.1 - 4.2.4 will reference the sensor names with associated metadata described in this section. The complete list of sensors is given in Table 4.2 and Table 4.3 and a sensor naming convention legend is given in Table 4.1 for clarity.

Table 4.1: Sensor name legend.

Short Name	Name	Short Name	Name
adc	air data computer	magn	3-axis magnetometer
baro	barometric altimeter	odom	odometer
cam	optical camera	prdr	pseudorange/delta range sensor
comp	magnetic compass	range	ranging sensor
gps	position and/or Velocity GPS	rfid	RFID device
incl	optical camera	sos	signal of opportunity sensor
ins	optical camera	ssign	stop sign sensor
laser2D	optical camera	step	step sensor
laser3D	optical camera	tra	terrain referenced altimeter

Table 4.2: Complete list of sensors used for all simulations with associated noise parameters.

Name	Wind STD (mi/hr)					
adc1	30					
Name	Baro Bias STD (m)	Baro Bias TC (s)	Quantization (m)			
baro1	2E-3	10	1			
baro2	2E-3	10	1			
Name	Meas STD ($unitless$)	Type				
cam1	1e-1	mono				
cam2	1e-1	mono				
cam3	1e-1	mono				
cam4	1e-1	mono				
cam5	1e-1	stereo				
Name	Comp Bias STD (deg)	Comp Bias TC (s)	Meas STD (deg)			
comp1	1	10	1			
comp2	1E-1	20	1E-1			

Table 4.3: Complete list of sensors used for all simulations with associated noise parameters.

Name	North Meas STD (m)	East Meas STD (m)	Down Meas STD (m)			
gps1	4	4	8			
Name	Incl Bias STD (deg)	Incl Bias TC (s)	Meas STD (deg)			
incl1	1	1	5E-1			
Name	Accel Bias STD (m/s^2)	Gyro Bias STD (rad/s)	Accel Bias T (s)	Gyro Bias T (s)	VRW $\frac{\frac{m}{s}}{\sqrt{s}}$	ARW $\frac{\frac{rad}{s}}{\sqrt{s}}$
ins1	1.96E-1	8.7E-3	3600	3600	4.3E-3	6.5E-4
ins2	1.96E-1	8.7E-3	3600	3600	4.3E-3	6.5E-4
ins3	1.96E-1	8.7E-3	3600	3600	4.3E-3	6.5E-4
ins4	9.8E-3	4.8481E-6	3600	3600	9.5E-3	8.73E-5
ins5	9.8E-3	4.8481E-6	3600	3600	9.5E-3	8.73E-5
ins6	2.45E-4	7.2722E-9	3600	3600	2.3833E-4	3.8178E-5
Name	x Meas STD (m)	y Meas STD (m)	ψ Meas STD (deg)			
laser2D1	5E-2	5E-2	1			
Name	$\boldsymbol{\zeta}$ Meas STD (m)					
laser3D1	5E-2					
Name	North Meas STD (m)	East Meas STD (m)	Down Meas STD (m)			
magn1	10	10	15			
Name	SF Bias STD	Meas STD				
odom1	1E-3	5E-2				
Name	Clk Err Bias STD (s)	Clk Err Bias TC (s)	Meas STD (m)			
prdr1	1E-7	10	1			
Name	Clk Err Bias STD (s)	Clk Err Bias TC (s)	Meas STD (m)			
range1	1E-8	10	1			
Name	N/A					
rfid1						
Name	North Pos STD (m)	East Pos STD (m)				
ssign1	3	3				
Name	Clk Err Bias STD (s)	Clk Err Bias TC (s)	Meas STD (s)			
sos1	1E-8	10	1E-8			
Name	Bias STD	Meas STD				
tra1	1E-2	1				

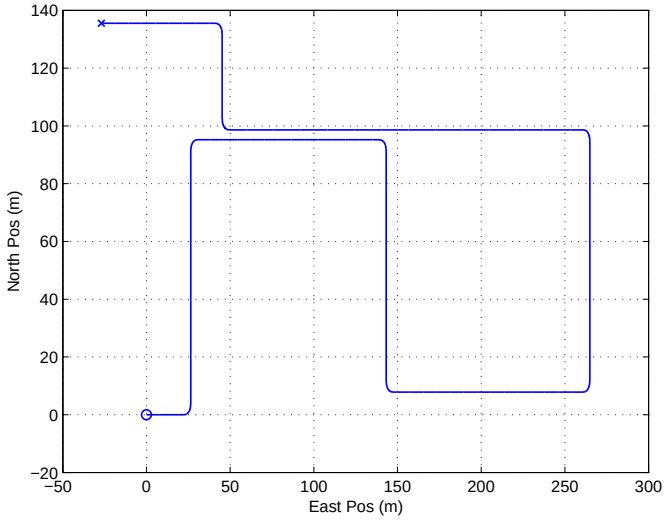

Figure 4.4: Top view of the pedestrian building path. The starting point is indicated by the circle and the stopping point by the 'x' mark.

4.2 Scenarios

Simulations were compiled for six separate scenarios. Two of the six scenarios pertain to a pedestrian, three for a ground vehicle, and one scenario for an aircraft. The first scenario for both the pedestrian and vehicle are short, straight paths designed to show the performance of subtler sensors added systematically. Alternate longer paths are generated for all three platforms to demonstrate sensor set navigation performance over larger amounts of time and in higher dynamic maneuvers. The aircraft is tested in only one very long scenario that includes launch, flight, and landing.

For each scenario, vehicle error states are given in plots. Plots in which no comparison is made with another sensor set are represented using dashed blue and black lines, where the mean is given by the blue line and associated standard deviation given by the darker black line. When comparisons are made on a single plot, the prior test that is compared against is displayed using the same blue and black line convention, and the current test is represented using lighter solid red and green lines

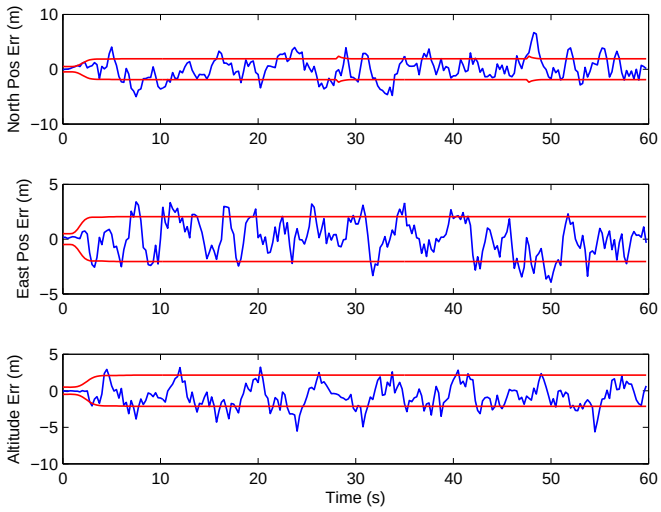

Figure 4.5: Pedestrian straight path position error with only *gps1* in operation. The error is given be the blue line and filter calculated standard deviation by the red lines.

for the mean and standard deviation respectively. For every plot the quantities are completely described, however the current test almost always includes all the previous test sensors plus one. Therefore, as a general rule of thumb, the lighter solid red and green lines will represent the equal to or better navigation solution.

4.2.1 Pedestrian. Tracks for the pedestrian were constructed to exhibit slow dynamics. The first path is straight and level where the platform quickly accelerates to a max speed, travels at a constant velocity, and the comes to a stop. The second pedestrian path is built to simulate a pedestrian traveling into, throughout, and then exiting a building.

4.2.1.1 Straight Path. The pedestrian straight path contains no attitude or elevation changes. The dynamics involve accelerating to a max speed of $3.5m/s$, maintaining constant speed and then coming to a stop. The duration of the run is one minute and the distance traveled is $180m$. The sensor set that was tested in the pedestrian straight path is given in Table 4.4. We begin our analysis of the

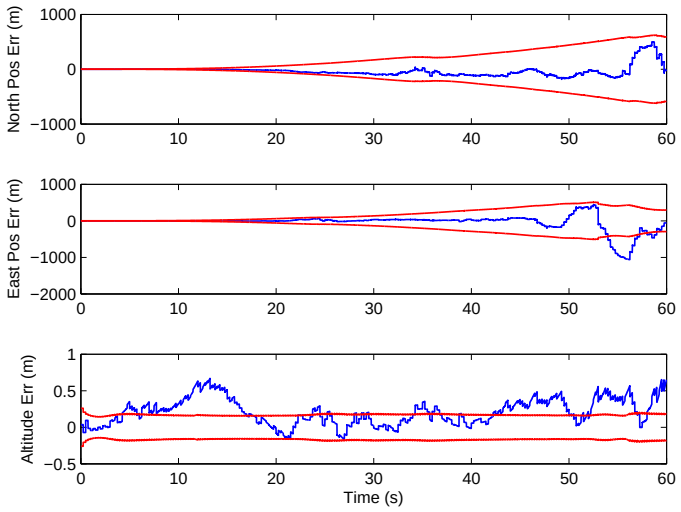

Figure 4.6: Pedestrian straight path position errors with only a MEMs grade IMU and barometric altimeter in operation

Table 4.4: Complete table of sensors for the pedestrian straight path

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
baro	1	4	ins	1	100
camera	1,5	4	incl	1	4
comp	1	4	prdr	1	1
gps	1	1	steps	1	N/A

pedestrian straight path using GPS only as a solution to compare other sensor sets against. The GPS only solution is given in Figure 4.5.

Now considering an environment with limited GPS signal, a navigation solution of the MEMs grade IMU with an aiding altitude provided by the barometric altimeter is shown in Figure 4.6. Due to the very IMU high bias drift and measurement noise, position errors in the north and east channels are very high. The effect of the altimeter on the altitude channel is very pronounced, not only because the drift errors can now be estimated and unknown errors in the normal gravity vector can be resolved.

We now consider bringing in an additional sensor. Often GPS positon solutions are unavailable when less than the required minimum of four satellites are visible. This can be the result of obstructing terrain, buildings, or other structures. Modeling an underdetermined GPS setup, it is desired to check the navigation ability using

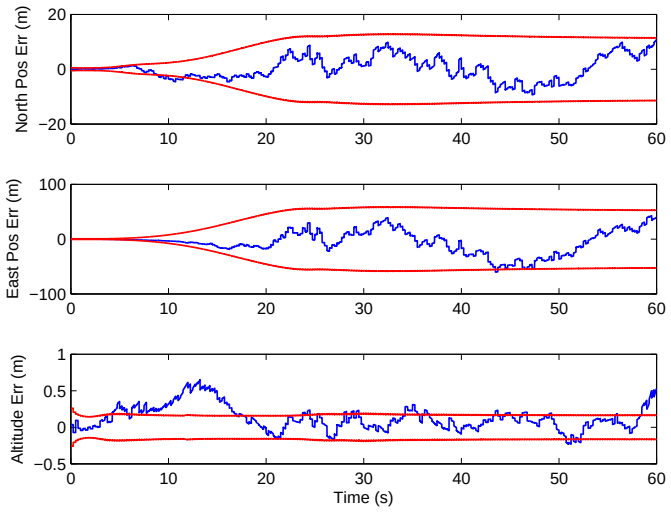

Figure 4.7: Pedestrian straight path position errors with the MEMs IMU, barometric altimeter, and two available pseudoranges in operation

less than for satellite pseudoranges. Using only two available pseudoranges with the GNSS pseudorange sensor, the errors in the east and north channels are significantly reduced as seen in Figure 4.7 compared to Figure 4.6. In fact, the error reduction is so large that the comparison requires separate plots for all errors to be visible. The error in the north channel has been reduced to a maximum of approximately $15m$ while the error in the east channel has a max error on the order of $50m$. The ability to use the two pseudoranges is due largely in part to the barometer which locks down the vertical channel and coupled with the IMU enables some estimation of the receiver clock error.

Limiting the range of motion, the addition of the step sensor effectively locks down the errors in the north and vertical channels. A comparison between the IMU, barometric altimeter, and GNSS pseudorange sensor integration with the addition of the step sensor is given in Figure 4.8. Modest improvements are obtained on the north channel, while the improvement on the east channel is so large that it again is difficult to view if compared on the same scale as the previous errors.

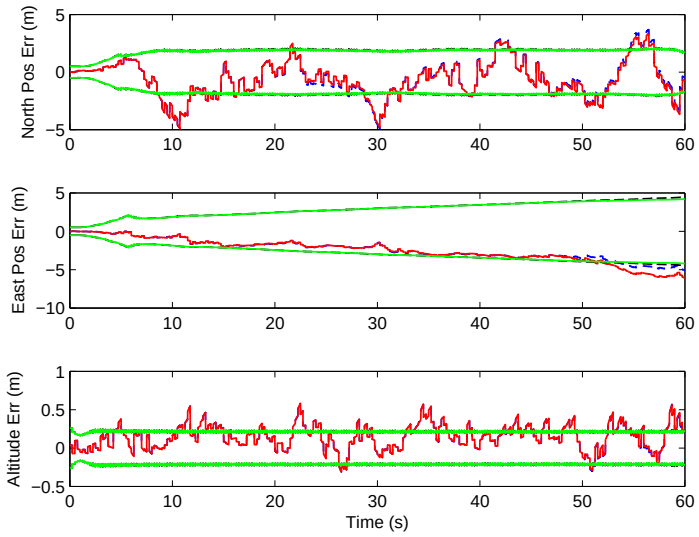

Figure 4.8: Pedestrian straight path position errors with the MEMs IMU, barometric altimeter, two available pseudoranges, and step sensor in operation compared with the addition of the monocular camera. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

The results of adding a monocular camera are examined. In the first comparison test on a single plot, the MEMs IMU, barometric altimeter, GNSS pseudorange sensor, and step sensor integration is compared with the addition of a monocular camera in Figure 4.8. The addition of the camera changes the position errors very little in the north and east channels, but provides improvement to east channel decreasing the error and bring it back to having a more zero mean characteristic, which was less noticeable with the previous addition of the step sensor.

We now consider the pedestrian attitude error. Because resolving the specific force due by gravity in the accelerometer measurements into the NED frame requires knowledge of the attitude, a dynamic connection is created between the attitude and platform positioning states. Because of this, each additional sensor that improved the position solution also "improved" the pedestrian attitude estimate. The reverse is also true as well, so the addition of attitude measuring sensors should improve the vehicle position estimate. Other than the IMU, the camera is the first sensor to

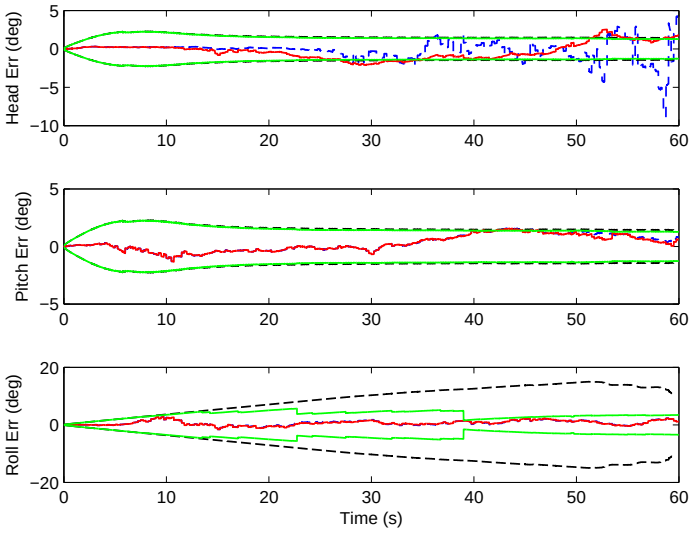

Figure 4.9: Pedestrian straight path attitude errors with the MEMs IMU, barometric altimeter, two available pseudoranges, and step sensor position error solution compared with the addition of a monocular camera. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

provide a measurement that is directly connected with the attitude. The pedestrian attitude comparison of the previous sensor set with the addition of the camera is given in Figure 4.9. In Figure 4.9, some of the improvement seen is a smoothing of the attitude error in the heading angle, no change in the pitch angle, and the certainty in the roll angle has be significantly lessened so that the STD now better characterizes the estimate.

Next the errors in the position solution are once again reduced by adding a compass and inclinometer. A comparison between the previous implementation with the IMU, barometric altimeter, GNSS pseudoranges sensor, step sensor, and monocular camera with the addition of the inclinometer and compass is shown in Figure 4.10. The standard deviation of each attitude channel has now become very low and somewhat close to exceeding 1σ STD line more than 42% of the time. Concerning the position error with the addition of the inclinometer and the compass there appears to be a smoothing of the altitude errors. As for the north and east channels, it is very

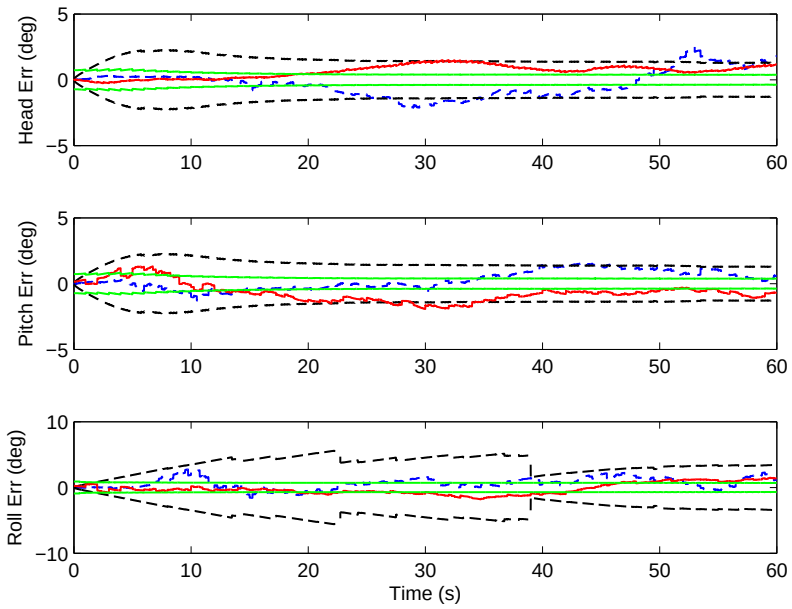

Figure 4.10: Pedestrian straight path attitude errors with the MEMs IMU, barometric altimeter, two available pseudoranges, a step sensor, and a monocular camera compared with the addition of a compass and an inclinometer. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

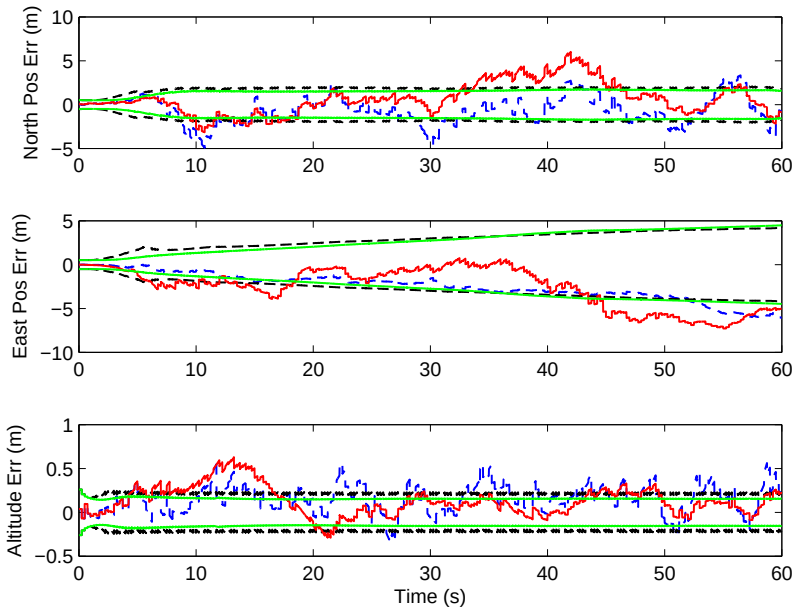

Figure 4.11: Pedestrian straight path position errors with the MEMs IMU, barometric altimeter, two available pseudoranges, a step sensor, and a monocular camera compared with the addition of a compass and an inclinometer. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

difficult to draw a conclusion whether the inclusion of the compass and inclinometer reduced the errors. What does appear to be happen in Figure 4.11 is the transition errors back to being zero mean white Gaussian. Overall this is then an improvement as the errors should appear this way.

Finally the complete sensor set is compared to the GPS position solution originally computed. This is given in Figure 4.12. Examining these results, it is seen that the combination of MEMs IMU, barometric altimeter, two available pseudoranges, the step sensor, a monocular camera, a compass, and an inclinometer are comparable to the GPS by itself in the north and east channels, and in the vertical channel is significantly better thanks to the barometric altimeter. In the scenario that follows this one, a longer more rigorous path is examined.

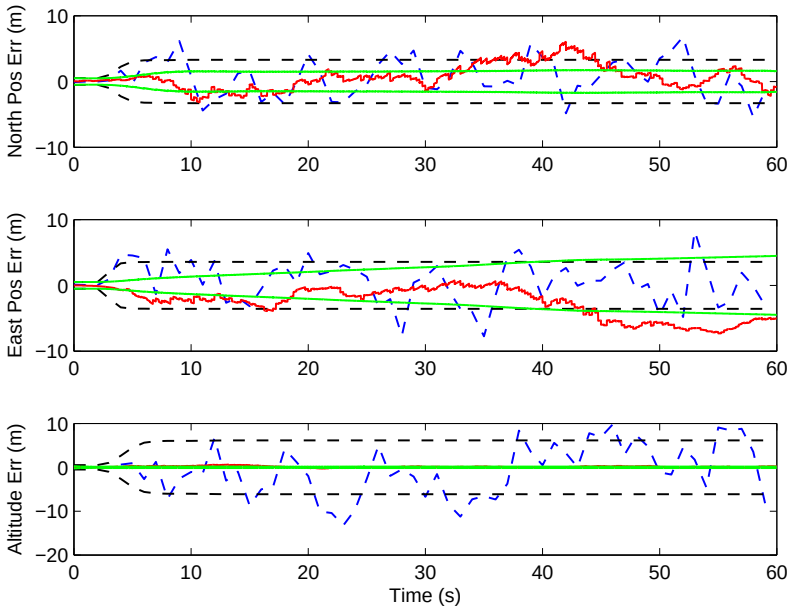

Figure 4.12: Pedestrian straight path position errors with a single GPS compared against the MEMs IMU, barometric altimeter, two available pseudoranges, an step sensor, and a monocular camera compared with the addition of a compass and an inclinometer sensor set. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

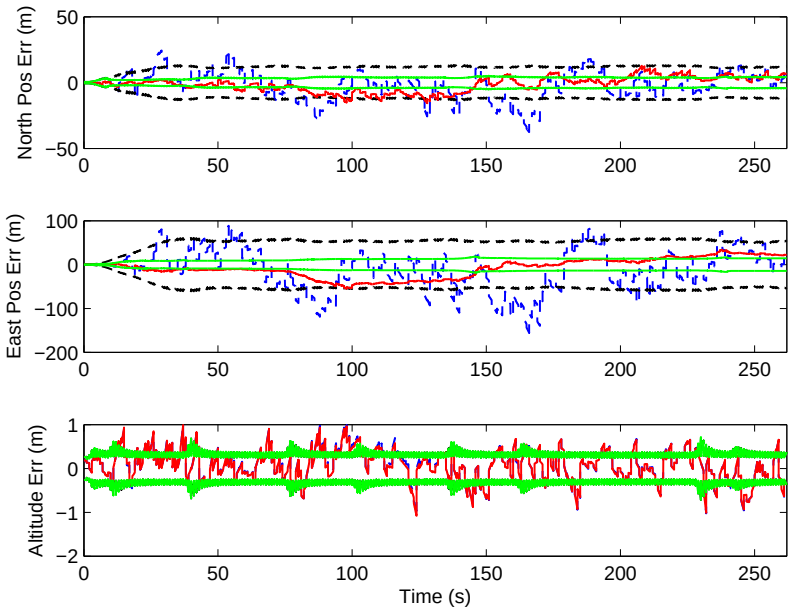

Figure 4.13: Pedestrian building path position errors with a single MEMs IMU, barometric altimeter, and two available pseudoranges compared with the addition of the step sensor. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

4.2.2 Building Path. While the straight path proved to be a useful diagnostic test for positioning sensors, it does not provide very valuable insight into the quality of the attitude estimation or position solution over long periods of time. Thus the pedestrian building path was designed to be the next test for the sensor integration. The pedestrian building path has a duration of over four minutes, covers a distance of over $850m$, and consists of many 90 degree turns and speed fluctuations. A top view of the pedestrian building path is given in Figure 4.4. The complete set of sensors used for this path are referenced in Table 4.5. Now considering the sensor set for this scenario, a single MEMs grade IMU, barometric altimeter, and two pseudorange measurements are employed for the first non-GPS positioning test. Integrating the measurements from the three sensors together, considerably good position tracking is obtained. Continuing the sensor addition process, an additional MEMs IMU is added to the sensor set. The comparison for the single MEMs IMU and the double MEMs

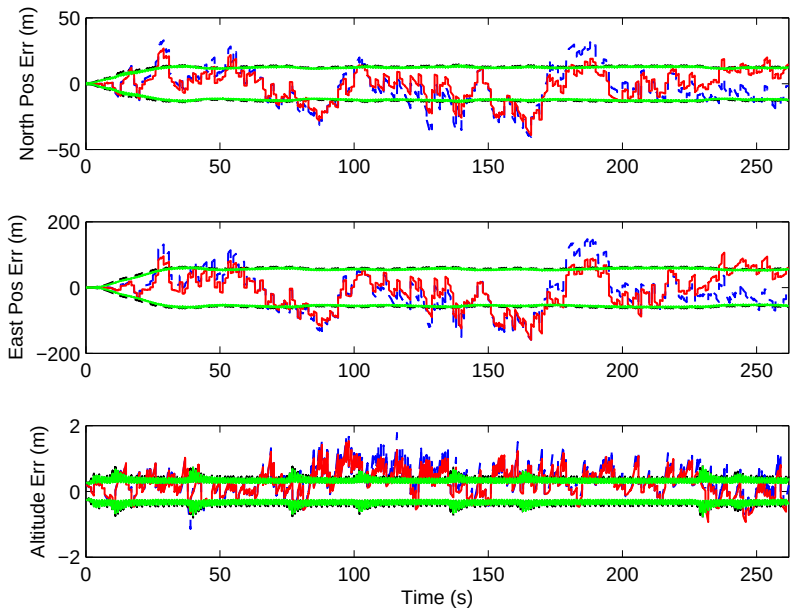

Figure 4.14: Pedestrian building path position errors with a single MEMs IMU, barometric altimeter, and two available pseudoranges compared with the inclusion of an additional MEMs IMU. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

Table 4.5: Table of sensors for the pedestrian building path.

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
baro	1	1	ins	1,2,3,4	100
camera	1,2,5	4	magn	1	0.1
comp	1	0.25	prdr	1	1
gps	1	4	step	1	N/A
incl	1	4			

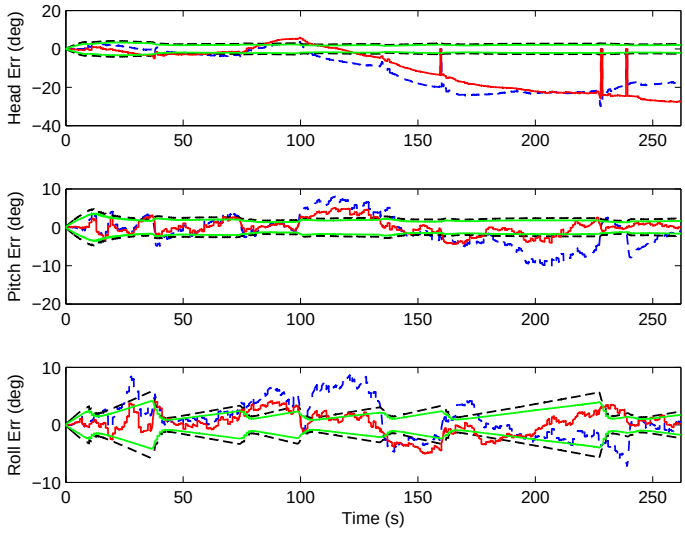

Figure 4.15: Pedestrian building path attitude errors with a single MEMs IMU, barometric altimeter, and two available pseudoranges compared with the addition of a second MEMs IMU, an inclinometer, and a compass. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

IMU implementation position error is given in Figure 4.14. Significant position error reduction is seen on the north and east channels, although some significant errors are observed on the east channel that are starting to dangerously exceed the state estimate STD.

Examining the attitude errors from the, seen in Figure 4.15, significant improvements are made on the roll and pitch states estimates due to the additional observability gained by the positioning sensors. The heading appears to have improved, however, estimates still suffers from large errors due to the fact that the only observability is provided by the IMUs. Similar results are noted with the addition of a third MEMs IMU with very little improvement to the position states seen Figure 4.16. The attitude remains mostly the same, with the exception of the a very large improvement seen on the heading estimate seen in Figure 4.17. Continuing the sensor integration, the step sensor, compass and inclinometer are added to the sensor array. Because the step sensor acts to constrain the range of motion, and the

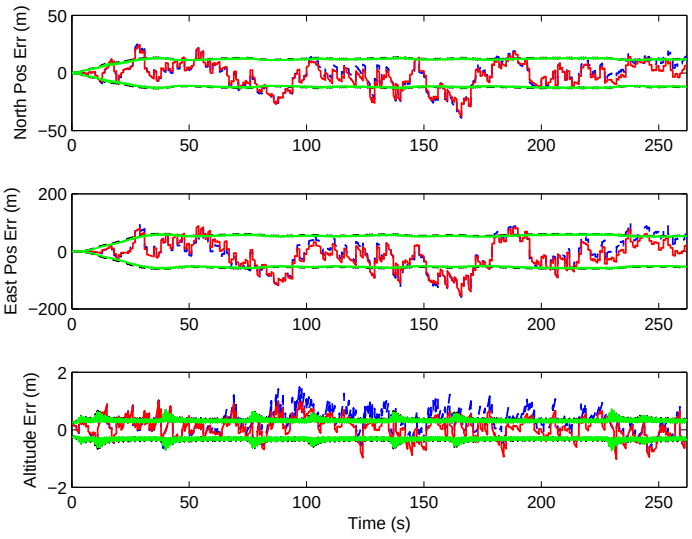

Figure 4.16: Pedestrian building path position errors with a single MEMs IMU, barometric altimeter, and two available pseudoranges compared with the addition of a MEMs IMU and an inclinometer sensor set. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

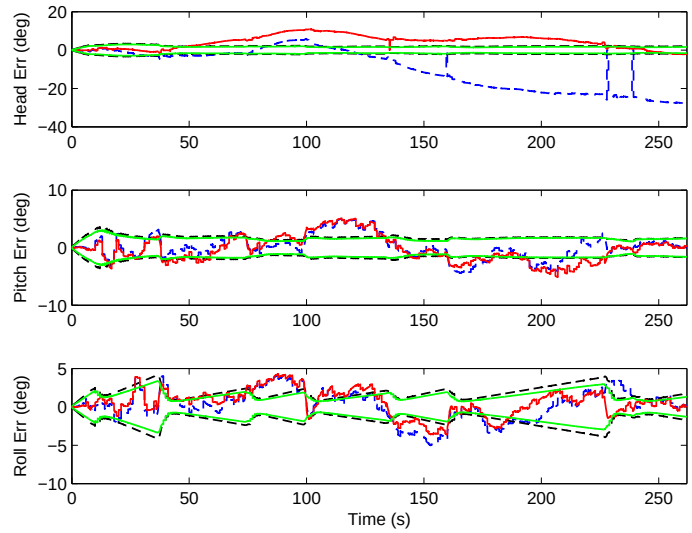

Figure 4.17: Pedestrian building path position errors with two MEMs IMUs, barometric altimeter, and two available pseudoranges compared with the addition of a third MEMs IMU, an inclinometer, and a compass. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

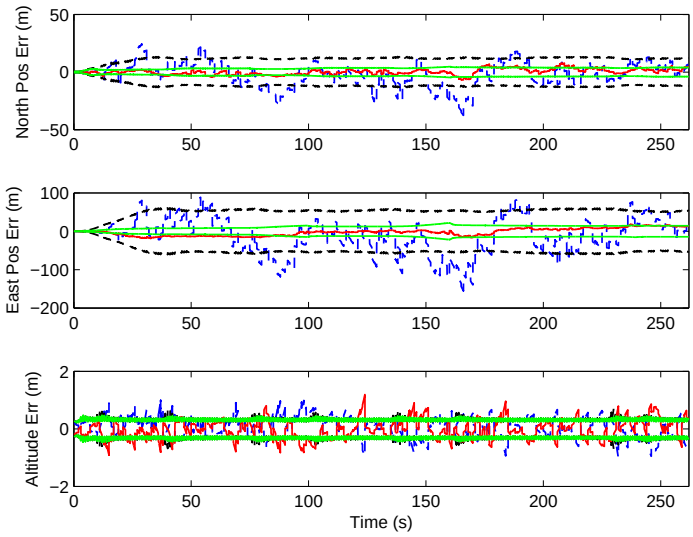

Figure 4.18: Pedestrian building path position errors with three MEMs IMUs, barometric altimeter, two available pseudoranges, step sensor, an inclinometer, and a compass. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

inclinometer and compass provide absolute roll, pitch, and heading estimates, the position solution is substantially improved and the position errors for the entire run are reduced to a max STD of $20m$ in the north channel, $10m$ in the east channel, and only $2m$ in the vertical directions as seen in Figure 4.18. The final sensor addition for the pedestrian path involves the addition of the 3-axis magnetometer. The 3-axis magnetometer provides full three dimensional position information every $10s$. Incorporating the $3-axis$ magnetometer into the filter gives results that are on the same level of accuracy as the GPS for the entire run as seen in the comparison in Figure 4.19.

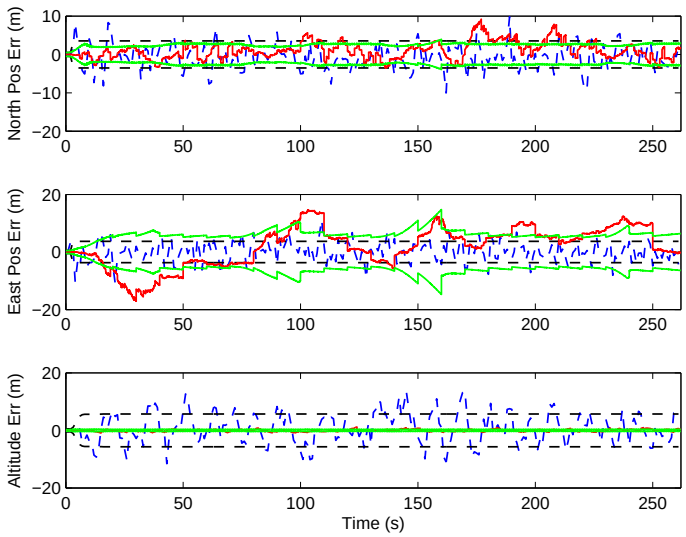

Figure 4.19: Pedestrian building path position errors with triple MEMs IMUs, barometric altimeter, two available pseudoranges, step sensor, an inclinometer, and a compass compared with the standalone GPS. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

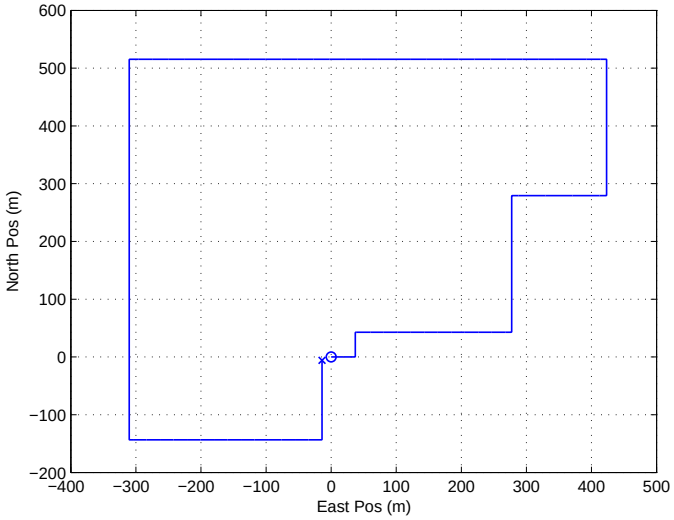

Figure 4.20: Top view of the vehicle city path. The starting point is indicated by the circle and the stopping point by the 'x' mark.

Table 4.6: Table of sensors for the vehicle straight path.

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
baro	1	1	ins	1,4	100
camera	1,2,5	4	laser2D	1	10
comp	1	2	prdr	1	1
gps	1	4	range	1	N/A
incl	1	4	odom	1	10

4.2.3 Vehicle.

4.2.3.1 Straight Path. The vehicle straight path is designed to test low observability sensors that are not used on the pedestrian. This includes the signal of opportunity sensor, ranging sensor, and the vehicle odometer. The path consists of a quick acceleration to $35mph$, traveling at a constant rate, and then slowing to a stop within one minute. During then run, the vehicle attitude and elevation are kept constant. The complete list of sensors used at some point in this path are given in Table 4.6. The duration of the vehicle straight path run is one minute and covers a distance of $780m$. In the first sensor setup, the vehicle position is tracked using the MEMs grade IMU, the barometric altimeter, the odometer, a compass, and two pseudoranges. Results obtained are very similar to those seen in the similar case

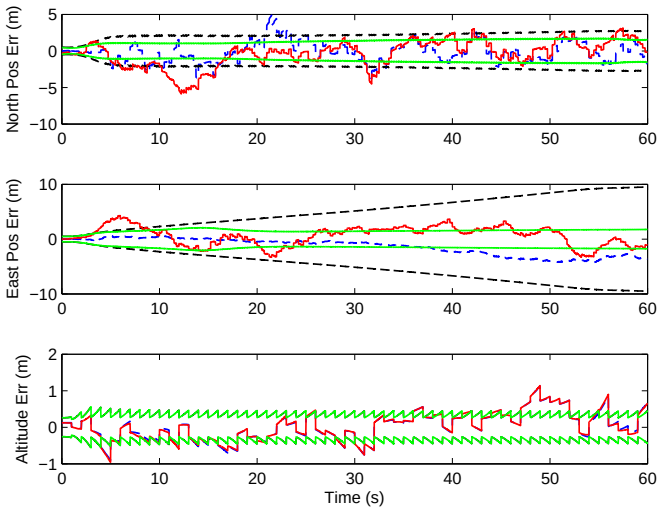

Figure 4.21: Vehicle straight path position errors with the MEMs IMU, barometric altimeter, two available pseudoranges, the odometer, and a compass compared with the addition of the ranging and the signal of opportunity sensors. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

with the step sensor and the pedestrian. Next the ranging and signal of opportunity sensors is added to the sensor array and the comparison is shown in Figure 4.21. The addition of the ranging and signal of opportunity sensors, provides the additional north and east information. In both the north and east channels, the position errors have become more white zero mean Gaussian and are much better characterized by their STD.

The final addition to the vehicle straight path is the 2D laser scanner. Due to numerical issues encountered when using the 2D laser scanner together with the odometer, the 2D laser scanner will replace the odometer for the next sensor set. These errors were due to the implementation of both sensors using the delayed-state equations, but this conflicting relationship was not observed between all delayed-state sensors. Adding in the 2D laser scanner, provided further improvements to the position even with the removal of the odometer. The most pronounced improvement

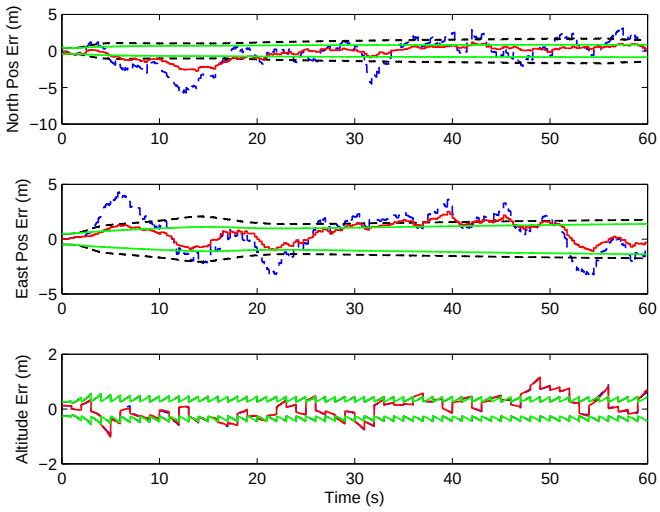

Figure 4.22: Vehicle straight path position errors with the MEMs IMU, barometric altimeter, two available pseudoranges, the odometer, and a compass compared with the same sensor setup but the odometer replaced by the 2D laser scanner. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

is seen on on the east position channel where the STD is reduced and characterizes the mean very well as seen in Figure 4.22.

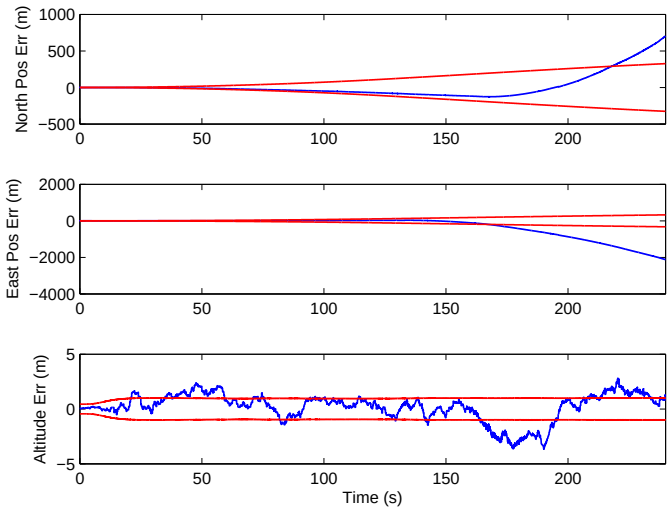

Figure 4.23: Vehicle urban path position errors with the MEMs IMU and barometric altimeter.

Table 4.7: Table of sensors for the vehicle urban path.

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
baro	2	1	magn	1	0.05
camera	1,2,3,4,5	4	odom	1	10
comp	1	4	prdr	1	1
gps	1	1	range	1	1
incl	1	4	rfid	1	N/A
ins	4	100	sos	1	1
laser2D	1	10	ssign	1	N/A
laser3D	1	10			

4.2.3.2 Urban Path. The urban path tests the multi sensor configurations in scenarios with many turns, relatively low dynamics, and provides an area for testing the stop sign sensor. The top view of the city path is given in Figure 4.20. Multiple 90 degree turns at a slow speeds of $15mph$ and less are taken through roadways similar to that seen in an urban neighborhood. During the longest stretch of the run, the vehicle accelerates to a top speed of $45mph$, while traveling down an incline. After this segment, the vehicle finishes the track driving $35mph$ until it reaches the neighborhood area again. There the vehicle slows to speeds of $15mph$ or less and eventually comes to a stop at approximately the same location as starting point. The vehicle city path has a duration over $240s$ and covers a distance of $1.5mi$.

Since this is the first time the tactical grade INS has been used, the complete path is first calculated using only the INS with an aiding altitude. As expected, the position errors start out small but soon grow very quickly, especially given the large number of turns in the urban path, which amplify errors in the position due to attitude errors unlike the case in the straight path. The INS and aiding altitude position errors are shown in Figure 4.23.

The vehicle path is then examined using the tactical IMU, two psuedoranges, 3-axis magnetometer, the 3D laser scanner, the odometer, two pseudoranges, and available RFID tags and compared to the standalone GPS solution. The resulting position errors are given in Figure 4.24. The majority of the position error reduction over the INS altimeter sensors integration is due largely in part to the 3D laser scanner. Some of the most severe errors seen in Figure 4.24 are due to the low observability of the 3D laser scanner with height, since the majority of identifiable planes in an urban environment have normals that are parallel to the ground, and few have normals orthogonal to the ground.

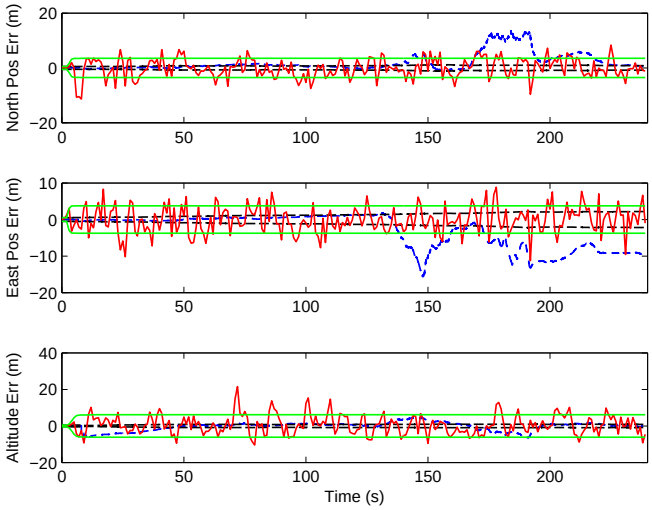

Figure 4.24: Vehicle urban path position errors with the tactical IMU, two psuedoranges, 3-axis magnetometer, the 3D laser scanner, the odometer, two pseudoranges, and available RFID tags compared to standalone GPS. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

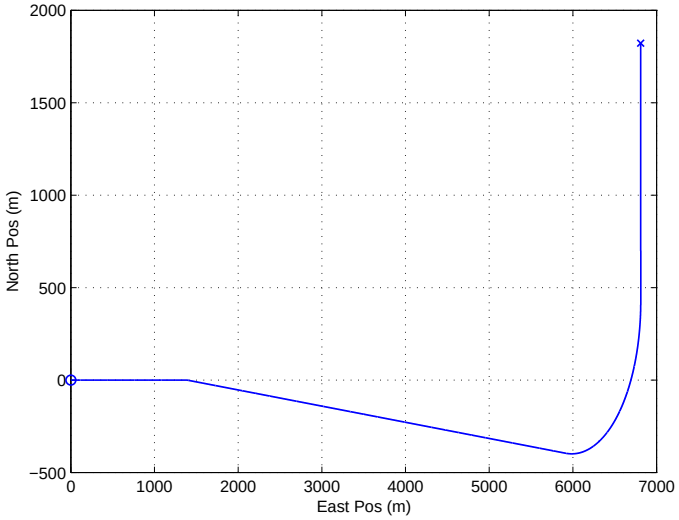

Figure 4.25: Top view of the vehicle interstate path. The starting point is indicated by the circle and the stopping point by the 'x' mark.

4.2.3.3 Interstate Path. The Interstate vehicle path allows analysis of the vehicle accelerating quickly onto an Interstate, traveling at high speed, changing elevation, making a wide sweeping turn that induces a roll angle, and then slowing to a stop. The scenario duration is approximately 4 minutes long and the path traversed covers over $5mi$. The top speed in this simulation is approximately $65mph$. The list of sensors selected from for this path are given in Table 4.8. The navigation in

Table 4.8: Table of sensors for the vehicle Interstate path.

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
baro	2	1	ins	1,4	100
camera	1,2,5	4	magn	1	0.05
comp	1	4	odom	1	10
gps	1	1	prdr	1	1
incl	1	4			

the vehicle Interstate path is tested using the barometric altimeter, tactical grade INS, a compass, an inclinometer, two pseudoranges, the odometer, and the 3-axis magnetometer. The resulting position errors are then compared against the GPS only solution in Figure 4.26. Just with the inclusion this sensor set, nearly the same results are obtained in the north position channel. In the east channel where the

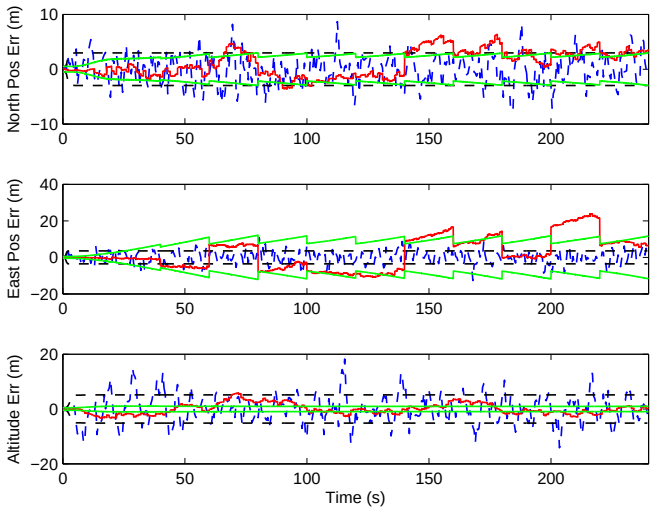

Figure 4.26: Vehicle Interstate path position errors with the barometric altimeter, tactical grade INS, a compass, an inclinometer, two pseudoranges, the odometer, and the 3-axis magnetometer compared with standalone GPS. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

majority of the motion takes place, less accuracy is noted with large non-GPS sensor set than with the standalone GPS. In the horizontal channel, greater accuracy is noted with the non-GPS sensor set even with the less accurate barometric altimeter chosen to represent errors due to velocity, and distance from pressure reference.

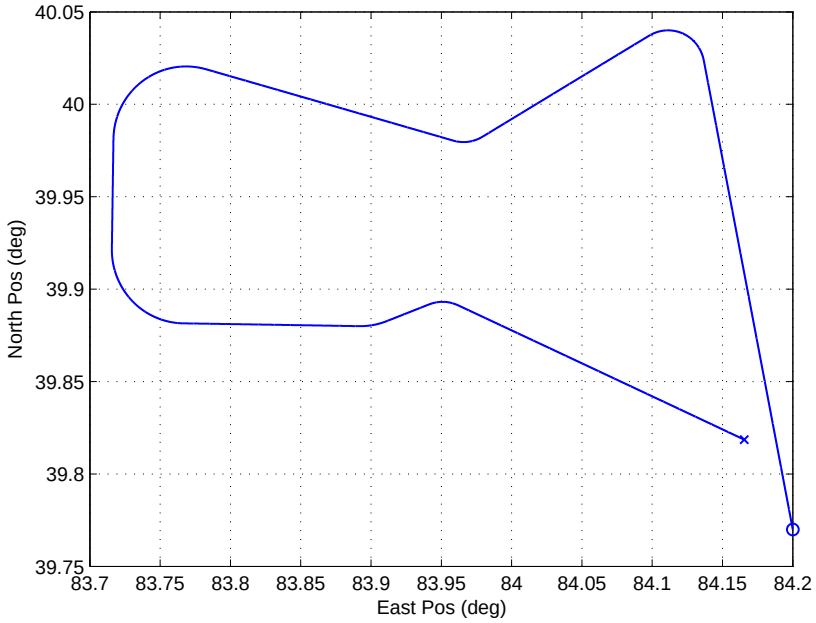

Figure 4.27: Top view of the San Gabriel aircraft path. The starting point is indicated by the circle and the stopping point by the 'x' mark.

Table 4.9: Table of sensors for the San Gabriel aircraft path.

Sensor Type	Sensor Id	Meas Rate (Hz)	Sensor Type	Sensor Id	Meas Rate (Hz)
adc	1	2	ins	6	100
baro	1	1	prdr	1	1
camera	1,2	4	range	1	1
comp	1	4	sos	1	1
gps	1	1	tra	1	1
incl	1	4			

4.2.4 Aircraft San Gabriel Flight. The aircraft simulation is a shortened version of a Profgen provided example of an aircraft trajectory over the California San Gabriel Mountains. The San Gabriel path includes launch, stable flight, and landing. The simulation duration is approximately 15 minutes, and the aircraft traverses over $71mi$ in it. The complete list of sensors selected from in this path are given in Table 4.9.

The navigation with the aircraft is first performed using only the navigation grade INS and the barometric altimeter. For the aircraft case, an altimeter with a higher altitude bias drift rate is considered to account for worse calibration errors

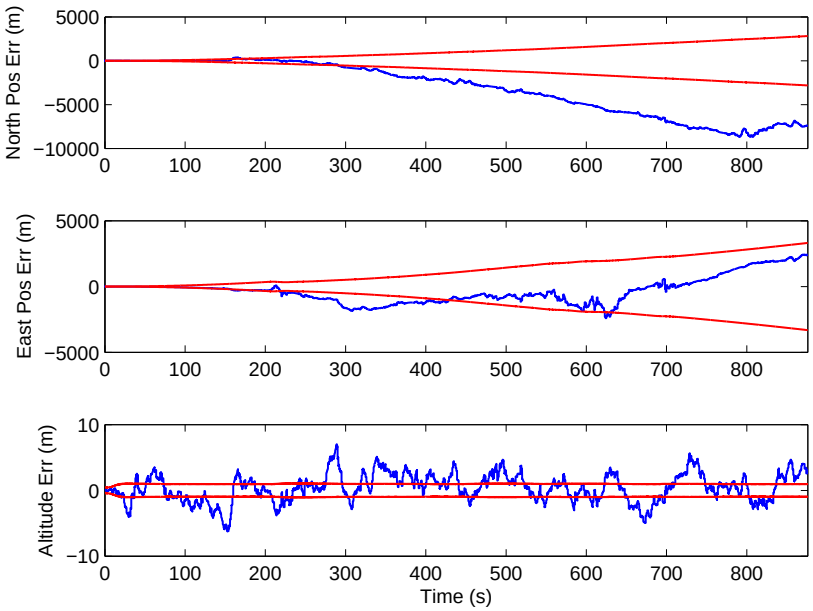

Figure 4.28: San Gabriel aircraft path position errors with the navigation grade INS and barometric altimeter.

and the increased distance from the pressure reference location. The navigation using only the INS and the altimeter maintains the aircraft's position for a good portion of the flight, however, without any over updates, the uncertainty and position errors in the north and east channels continue to grow as seen in Figure 4.28. The aircraft altitude tracking errors are bounded by the altimeter.

Next the ADC is added to the navigation sensor set. Due to the high accuracy of the navigation grade INS, the ADC has very little noticeable effect on the navigation solution when viewed only with the INS and barometer. The effect of the air data computer is best seen using the GPS that provides widely spaced measurements and a compass. In Figure 4.29, $0.1Hz$ GPS and compass position errors are compared against the added ADC for the first five minutes of the flight. The ADC significantly lowers the STD in the north and east channels which still completely characterize the error. As a final run, the position of the aircraft is tracked using the INS, ADC, barometric altimeter, two available pseudoranges, and the TRA. As shown in Figure 4.30, very good position accuracy is obtained using this setup. The position errors

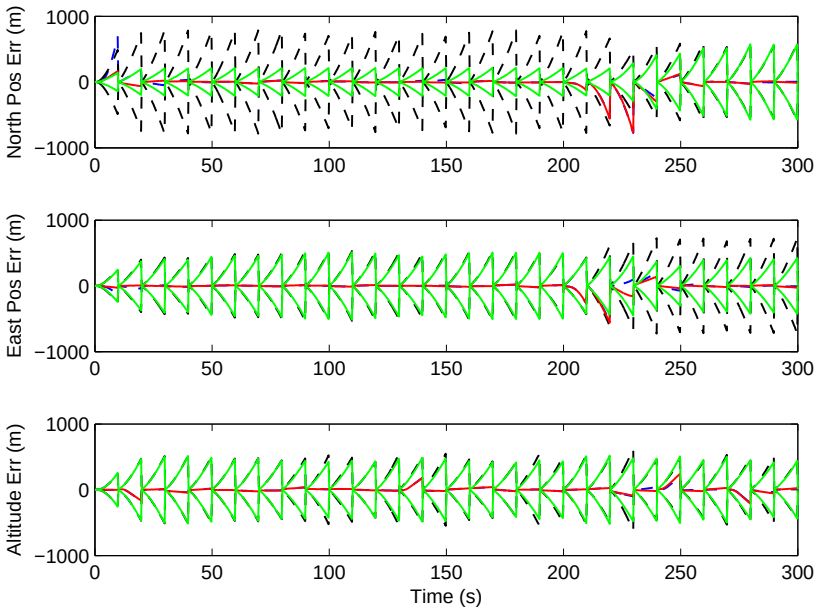

Figure 4.29: San Gabriel aircraft path position errors with the GPS and a compass compared with the addition of the ADC. The first described test mean and STD are displayed with the darker dashed blue and black lines, and the ensuing test mean and STD are given with lighter solid red and green lines.

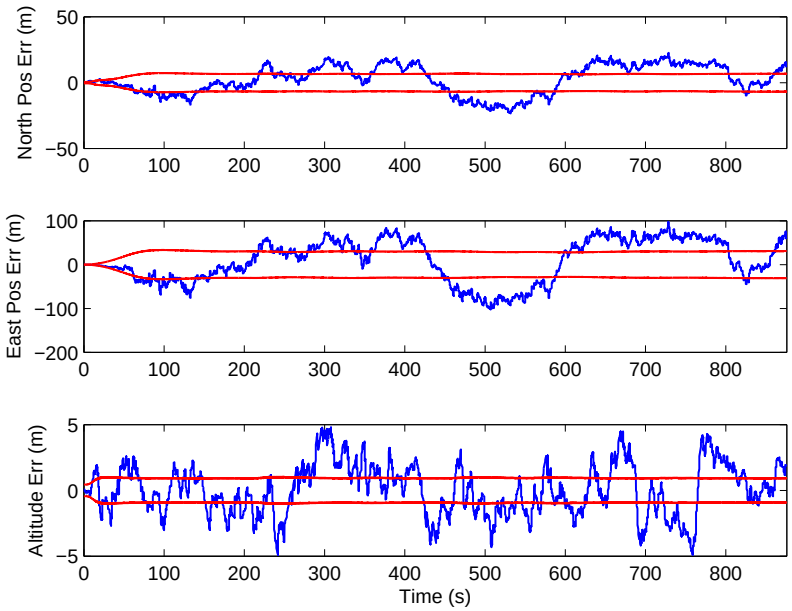

Figure 4.30: San Gabriel aircraft path position errors with the INS, ADC, barometric altimeter, two available pseudoranges, and the TRA.

in the north channel reach a maximum value of $20m$ and the position errors in the east channel have a maximum error of $70m$. The altitude maintains a very accurate estimate using the barometric altimeter with the TRA. All three position states are well characterized by their mean. Though the position tracking with the INS, ADC, barometric altimeter, two available pseudoranges, and the TRA was very good, the errors are still too large to compare directly with the standalone GPS on the same plots.

4.3 Summary

In order to produce the results in this thesis, trajectory paths had to be created for each of the three platforms. Six paths were created to test the varied sensor set under multiple dynamics and platform maneuvers. The software used to create the trajectory paths was Profgen Tools. In addition to navigation states, measurements had to be simulated for every sensor. Many sensors provide measurements that require more than just sampled states to represent, such as features and planar surfaces.

This measurements were simulated using custom code. Finally the sensor sets were defined, and the EKF filter was run for each of the simulated paths using multiple sensor configurations and the results generated were presented.

The results presented in this chapter show that for at least short paths, position tracking with accuracy levels similar to GPS are possible in a GPS-limited environment. Often a staple in the estimating the navigation solution was the inclusion of the pseudorange sensor. Though the GPS solution was by far underdetermined, the pseudoranges provided absolute positioning that was affected only by one error state because of the common receiver. Because other range sensors such as the signal of opportunity sensor or ranging sensor have separate clock errors, their benefit is much more limited than pseudoranges. Another suprisingly useful sensor was the step sensor and odometer. Matched with an IMU, pseudoranges, and the barometric altimeter, a very functional navigation system was developed that maintained position accuracy for relatively long periods of time.

V. Conclusion and Summary

The results and final conclusions are drawn in this chapter of this thesis. This includes discussion of the advantages of the implementations taken, what worked well, what did not work well, and the lessons that were learned.

5.1 Implementation Discussion

Many design choices were made early on in this research, such as the desired filter for sensors integration and sensor measurement models. The rationale behind these design decisions is discussed in this section giving the potential advantages and disadvantages. Lastly these decisions are evaluated now that the project is complete.

5.1.1 EKF Implementation. The main advantage of using the EKF in this thesis was the mass of readily available information concerning its implementation, examples of use, and known pitfalls. The EKF is ubiquitous in navigation research. This often makes its implementation desirable even though more accurate methods are available. In addition, use of the EKF was an ASPN program constraint.

Unfortunately, there are many disadvantages of using the EKF as well. Implementation of the 19 sensors discussed in this thesis was anything but trivial using an EKF. Building the Jacobian matrix for many measurement updates equations became a long tedious process. Blindly forming the Jacobian matrices of measurement update equations with respect to all states that appeared in the expression, as illustrated by EKF theory, often led to problems. This was due to non-observability in the measurement with respect to all states appearing in the update equation. Deviations from the normal formation of the Jacobian matrices were required to prevent filter divergence. This required considering the type of measurement, and whether or not any real observability was available of the state appearing in the measurement equation and how critical that quantity was to relating the measurement to other states.

The EKF implemented in traditional form also encountered problems from sensors that provide measurements that are correlated with estimated states such as the

terrain based altimeter or stop sign sensors. In fact, inclusion of the stop sign sensor in this project was partially performed to test and even cause the EKF solution to diverge. Incorporating these devices demonstrate the requirement for more intensive nonlinear filtering implementations as discussed in [13].

5.1.2 Sensors and Sensor Models. Many of the sensors used in this research provided a small but substantial impact on the navigation solution, at least given the very simple platform dynamics models described in Chapter III. Inclusion of ranging measurements such as GNSS pseudoranges, time difference of arrival, and beacon ranging helped to slow INS drift and lower position covariance. The stop sign sensor provided little useful information as implemented, and often brought more problems to the EKF than it solved due to its dependance on the current position and high likelihood of being matched to an incorrect stop sign. Distance measurements given by the step sensors for the pedestrian, and the odometer for the vehicle provided only a small amount of information by themselves, but matched with and underdetermined GPS scenario and an altimeter provided excellent aiding to a MEMs IMU.

The dominating sensors of the group provided substantial results. The compasses and inclinometers provided excellent aiding for attitude estimation and together almost completely encompass the very limited methods for bounding IMU attitude drift cheaply and effectively. Barometric altimeters provided essential support for the IMU by removing uncertainty in the normal gravity vector, and alignment.

In the case of the aircraft only, the air data computer actual air speed measurement provided a very useful method of navigation once the wind velocity vector was determined. Wind velocity determination requires some alternate measurement source such as GPS provided position of radar based measurements coupled with turning maneuvers described in [7]. Once an initial estimate of the wind velocity was determined, the air speed measurement coupled with the IMU provides an excellent combination. The TRA combined with a DTED map file provides an excellent source of altitude updates. Because this measurement is very dependent on the position

states, it can be a source for problems with the filter. This error is mostly accounted for by sampling the terrain heights over the entire position uncertainty region and calculating a mean and covariance of the region for the update. The validity of this method assumes correct modeling of the positions states plus it is very dependent on the assumption that the terrain height may be accurately modeled as Gaussian.

Camera sensors provide an excellent source for the ground platforms for bounding IMU drift errors, but in practice require a great deal of support. The camera projection model is anything but plug and play and requires relatively good position, attitude, and a well tuned model. Key to the successful camera implementation is an abundance of features available at the beginning of navigation where the initial position is known (even if only relatively so). The beginning of navigation is a broad term in this context, but ultimately is dependent on the quality of the IMU being used with the camera. When used together, the camera aids navigation by bounding the IMU drift errors. Over time, as position errors add up, the camera loses its ability to accurately locate new features and quickly can cause problems with the EKF, unless some form of absolute positioning becomes available.

The 2D laser scanners provided another excellent source of binding IMU errors and with the addition of the barometric altimeter, compass, and inclinometer can provide a complete navigational system for the low dynamics case, as long as there are an adequate number of objects for the scanner to base measurements from. The 3D laser scanner can provide complete platform navigation observability in urban environments combined with only an IMU. As implemented in this thesis, the 3D scanner could operate by itself to some degree, but this is unrealistic since an IMU is required in real data to correct for motion error and plane matching when relatively low dynamics specifications are exceeded [26]. Ultimately though, the 3D scanner provided one of best implementations of all the sensors, but as used in this project is dependent on urban environment structures.

5.2 Future Research

Due to the size of this project, there are many areas for improvement in future. The areas discussed in this section are simulation improvement, more detailed sensor models, and sensor fusion for a new sensor.

5.2.1 Realistic Simulations. Many of the sensors discussed in this thesis operate well under certain conditions, however scenarios can be devised that severely affect sensor operation. For example a compass works very well when placed upon a flat surface, but the heading measurement quickly degrades with tilt. Improved simulations of sensors would link the tilt of the platform to the amount of error seen in the compass measurement. Likewise, the main source of error with inclinometers is its changing motion. Measurement error in simulation would represent real data much more closely if the platform accelerations and angular rates were linked.

Including both types of these error would require detailed information to be known about the sensors in question. For simulations in this thesis, none of errors were specifically taken into account, but rather every measurement was corrupted using its exact model. That is, the compass TCB was simulated with a TCB that was independent of the platform tilt. More rigorous tests where errors that are coupled with the main sources seen in real data would be appropriate for future work.

5.2.2 Improved Sensor Models. Many of the sensor models used in this research are crude in nature. That is, many models had to be overly simplified in order to make a project of this size manageable in the constricted time frame. More detailed measurement models including other errors would provide a relatively simple and straight forward update to this work without requiring major code modification. The main source of difficulty would be the additional attention to physical sensor parameters.

As an example, the tilt induced errors previously described can be minimized if information about the tilt and its relationship to the compass error is known. This

issue is addressed in literature [18]. In this thesis, the induced error is never actually related to the roll and pitch states, but only accounted for using a TCB that is only related to the measurement. Information about the bias must be interpolated from the filter estimate of what the heading measurement should be. A simple relationship could be included if the measurement covariance was increased by being linked to the platform tilt states. More detailed approaches would also involve correcting the measurement if enough information about the sensor was known. Likewise a more representative measurement model of the inclinometer could be developed, where the bias is dynamically coupled with the acceleration and angular rate states.

5.2.3 Alternate Camera Implementation. As used in this project, the main benefit of stereo vision that made it superior to single camera vision was its ability to estimate the distance to a feature, by estimating the intersection of the camera feature projection vectors. This ability is not taken advantage of after initialization, but could be used in a method very similar to that of the 3D laser scanner for estimating position change between images. This could be implemented by forming the complete projection vector to the feature at each measurement. The actual camera model could then be modeled with relative positioning that does not possess such strongly correlated errors with the estimate position and attitude states and would be implemented as a delayed-state sensor. This implementation would not require feature estimation which would significantly lower computational burden.

An alternate form of navigation could be implemented with the monocular camera sensors as well similar to the stop sign sensor. By cataloging objects that could be recognized, absolute position could be gained using only an monocular camera coupled with an object detection. This would require considerable overhead however as the objects must be placed ahead of time and kept track of. A similar approach is taken in this project with the RFID tags, but with the camera, attitude information could also be gained using the same camera model employed in this project.

5.2.4 Detailed Platform Models. Higher detail platform models would provide valuable insight into this research. A simple improvement would involve constraining motion in each platform to the body forward and reverse axis. This would significantly improve the utility of sensors that provide low observability in higher dimensions such as odometers, step sensors, and ranging devices. More complicated improvements could be made by constructing detailed models that considered the physical characteristics of the platforms. These models could even display adaptive behavior such as seen in [21] where during times when there is high certainty of the platform navigation, the platform model parameters can be estimated.

5.2.5 Alternate Filter Implementations. Many options exist for integrating all the sensors described in this thesis. As previously described, the EKF is far from a perfect implementation and many other filters such as the unscented Kalman filter (UKF) and particle filter are widely considered more accurate. This is especially true with the particle filter for cases where estimated quantities or measurements are highly nonlinear and/or whose errors are not accurately described by a Gaussian distribution. Using the UKF would certainly simplify implementation by avoiding linearization of equations. Use of the UKF is considered superior to the EKF and even rivals the particle filter in terms of accuracy with a computation time on the same level as that of the EKF [32].

5.2.6 Camera/ 3D Laser Fusion. In a project with this many sensors, it might seem odd to introduce another sensor to the list for future discussion, however the camera and 3D laser scanner provide an intriguing sensor combination that minimizes many of the faults observed in the sensors individually. As described in previous research, the camera and 3D laser pair would provide an excellent new addition to the sensor list for this massive project.

Bibliography

1. Jared B. Bancroft and Grard Lachapelle. Data fusion algorithms for multiple inertial measurement units. *Sensors*, 11(7):6771–6798, 2011.

2. D. Bates. Navigation using optical tracking of objects at unknown locations. Master's thesis, Ohio University, 2007.

3. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

4. K.R. Britting. *Inertial Navigation Systems Analysis*. GNSS technology and applications series. Artech House, 2010.

5. R.G. Brown and P.Y.C. Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Wiley, 1997.

6. DMA WGS-84 Development Committee. Department of defense world geodetic system 1984 - its definition and relationships with local geodetic systems. In-House Technical Report 8350.2, Defense Mapping Agency, Washington, DC, September 1987.

7. D. Delahaye and S. Puechmorel. Tas and wind estimation from radar data. In *Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th*, pages 2.B.5–1 –2.B.5–16, oct. 2009.

8. Bon A. DeWitt and Paul R. Wolf. *Elements of Photogrammetry(with Applications in GIS)*. McGraw-Hill Higher Education, 3rd edition, 2000.

9. D. Eberly. *3D game engine design: a practical approach to real-time computer graphics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

10. J.A. Farrell, T.D. Givargis, and M.J. Barth. Real-time differential carrier phase gps-aided ins. *IEEE Transactions on Control Systems Technology*, 8(4):709 –721, jul 2000.

11. B. Friedland. Analysis strapdown navigation using quaternions. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(5):764 –768, sept. 1978.

12. Michael Hoffman. Schwartz warns against dependence on gps. `http://www.airforcetimes.com/news/2010/01/airforce_schwartz_012310/`. [Online; accessed 1-Feb-2012].

13. L. Hostetler and R. Andreas. Nonlinear kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control*, 28(3):315 – 323, mar 1983.

14. S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *The 11th Int. Symp. on AerospaceDefence Sensing, Simulation and Controls*, 1997.

15. S.P. Karatsinides. Enhancing filter robustness in cascaded gps-ins integrations. *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1001 –1008, oct 1994.

16. D. Krys and H. Najjaran. Ins assisted vision-based localization in unstructured environments. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3485 –3490, oct. 2008.

17. D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150 –1157, 1999.

18. S. Mangan, J. Wang, and Q.H. Wu. Measurement of the road gradient using an inclinometer mounted on a moving vehicle. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 80 – 85, 2002.

19. Peter S. Maybeck. *Stochastic Models, Estimation and Control, Vol II.* Academic Press, Inc., Orlando, FL, 1979.

20. Peter S. Maybeck. *Stochastic Models, Estimation and Control, Vol I.* Academic Press, Inc., Orlando, FL, 1994.

21. S.L. Miller, B. Youngberg, A. Millie, P. Schweizer, and J.C. Gerdes. Calculating longitudinal wheel slip and tire parameters using gps velocity. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 1800 –1805 vol.3, 2001.

22. Stanton H. Musick. *User's Guide For Profgen, A Trajectory Generator.* Air Force Research Laboratory, Wright-Patterson AFB, Ohio, December 2004.

23. H. Sairo, D. Akopian, and J. Takala. Weighted dilution of precision as quality measure in satellite positioning. *IEE Proceedings - Radar, Sonar and Navigation*, 150(6):430 – 436, dec. 2003.

24. S. Schneider, M. Himmelsbach, T. Luettel, and H.-J. Wuensche. Fusing vision and lidar - synchronization, correction and occlusion reasoning. In *2010 IEEE Intelligent Vehicles Symposium (IV)*, pages 388 –393, june 2010.

25. H. Jr Shumate. The radius of curvature in the prime vertical. *ITEA*, 30:159 – 163, 2009.

26. A. Soloviev and M.U. de Haag. Three-dimensional navigation with scanning ladars: Concept amp; initial verification. *IEEE Transactions on Aerospace and Electronic Systems*, 46(1):14 –31, jan. 2010.

27. James Stewart. *Multivariable Calculus Concepts and Contexts.* Thomson Higher Education, Belmont, CA, third edition, 2005.

28. Dan Sunday. Distance between lines and segments with their closest point of approach. `http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm`. [Online; accessed 15-Feb-2012].

29. David H. Titterton and John L. Weston. *Strapdown Inertial Technology*. The Institution of Electrical Engineers, Stevenage, Herts, UK, 2004.

30. C. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395 – 404, jun 1978.

31. Michael J. Veth. *Fusion of Imaging and Inertial Sensors for Navigation*. Ph.D. dissertation, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, August 2006. AFIT/DS/ENG/06-09.

32. Eric A. Wan and Rudolph van der Merwe. *The Unscented Kalman Filter*, pages 221–280. John Wiley & Sons, Inc., 2002.

33. R. Want. An introduction to rfid technology. *Pervasive Computing, IEEE*, 5(1):25 – 33, jan.-march 2006.

34. Eric W. Weisstein. "point-plane distance." from mathworld–a wolfram web resource. `http://mathworld.wolfram.com/Point-PlaneDistance.html`. [Online; accessed 1-Feb-2012].

35. R.K. Wells and A. Massey. A comparison of gps and doppler shift derived velocities. In *OCEANS '92. 'Mastering the Oceans Through Technology'. Proceedings.*, volume 2, page 607, oct 1992.

36. Li Xisheng, Kang Ruiqing, Shu Xiongying, and Yu Guanghua. Tilt-induced-error compensation for 2-axis magnetic compass with 2-axis accelerometer. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 3, pages 122 –125, 31 2009-april 2 2009.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
22-03-2012	Master's Thesis	Aug 2010 - Mar 2012

4. TITLE AND SUBTITLE

All Source Sensor Integration Using and Extended Kalman Filter

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Penn, Timothy R., 2d Lt, USAF

5d. PROJECT NUMBER
12G602

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/12-32

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Agency
Dr. Stephanie Tomkins
DARPA Strategic Technology Office (STO)
3701 N. Fairfax Drive - 3rd Floor
Arlington, VA 22203
(703) 248-1540, Stefanie.Tomkins@darpa.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
DARPA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The global positioning system (GPS) has become an ubiquitous source for navigation in the modern age, especially since the removal of selective availability at the beginning of this century. The utility of the GPS is unmatched, however GPS is not available in all environments. Heavy reliance on GPS for navigation makes the warfighter increasingly vulnerability as modern warfare continues to evolve. This research provides a method for incorporating measurements from a massive variety of sensors to mitigate GPS dependence. The result is the integration of sensor sets that encompass those examined in recent literature as well as some custom navigation devices. A full-state extended Kalman filter is developed and implemented, accommodating the requirements of the varied sensor sets and scenarios. Some 19 types of sensors are used in multiple quantities including inertial measurement units, single cameras and stereo pairs, 2D and 3D laser scanners, altimeters, 3-axis magnetometers, heading sensors, inclinometers, a stop sign sensor, an odometer, a step sensor, a ranging device, a signal of opportunity sensor, global navigation satellite system sensors, an air data computer, and radio frequency identification devices. Simulation data for all sensors was generated to test filter performance. Additionally, real data was collected and processed from an aircraft, ground vehicles, and a pedestrian. Measurement equations are developed to relate sensor measurements to the navigation states. Each sensor measurement is incorporated into the filter using the Kalman filter measurement update equations. Measurement types are segregated based on whether they observe instantaneous or accumulated state information. Accumulated state measurements are incorporated using delayed-state update equations. All other measurements are incorporated using the numerically robust UD update equations. Simulation results show the expected performance of improved navigation state estimation over time with the systematic addition of each sensor.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: UU			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Dr. John F. Raquet (ENG)
U	**U**	**U**	**UU**	151	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x:4580 john.raquet@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18